AudioCodes Mediant™ Media Gateways

SIP

Mediant 1000 & Mediant 600

User's Manual

Version 6.0

Document #: LTRT-83306

March 2010

Table of Contents

1 Overview 21

1.1 Mediant 1000 21
1.2 Mediant 600 22
1.3 SIP Overview 24

2 Configuration Concepts 25

3 Web-Based Management 27

3.1 Getting Acquainted with the Web Interface 28
3.1.1 Computer Requirements 28
3.1.2 Accessing the Web Interface 28
3.1.3 Areas of the GUI 30
3.1.4 Toolbar 31
3.1.5 Navigation Tree 32
3.1.5.1 Displaying Navigation Tree in Basic and Full View 33
3.1.5.2 Showing / Hiding the Navigation Pane 34
3.1.6 Working with Configuration Pages 34
3.1.6.1 Accessing Pages 34
3.1.6.2 Viewing Parameters 35
3.1.6.3 Modifying and Saving Parameters 38
3.1.6.4 Entering Phone Numbers 39
3.1.6.5 Working with Tables 39
3.1.7 Searching for Configuration Parameters 41
3.1.8 Working with Scenarios 42
3.1.8.1 Creating a Scenario 43
3.1.8.2 Accessing a Scenario 45
3.1.8.3 Editing a Scenario 46
3.1.8.4 Saving a Scenario to a PC 47
3.1.8.5 Loading a Scenario to the Device 48
3.1.8.6 Deleting a Scenario 49
3.1.8.7 Exiting Scenario Mode 50
3.1.9 Creating a Login Welcome Message 51
3.1.10 Getting Help 52
3.1.11 Logging Off the Web Interface 53
3.2 Using the Home Page 54
3.2.1 Assigning a Port Name 56
3.2.2 Resetting an Analog Channel 56
3.2.3 Viewing Analog Port Information 57
3.2.4 Viewing Trunks' Channels 58
3.2.5 Replacing Modules 59
3.3 Configuration Tab 60
3.3.1 Network Settings 60
3.3.1.1 Configuring the Multiple Interface Table 61
3.3.1.2 Configuring the Application Settings 66
3.3.1.3 Configuring the NFS Settings 67
3.3.1.4 Configuring the IP Routing Table 68
3.3.1.5 Configuring the QoS Settings 70
3.3.2 Media Settings 71
3.3.2.1 Configuring the Voice Settings 72
3.3.2.2 Configuring the Fax/Modem/CID Settings 73
3.3.2.3 Configuring the RTP/RTCP Settings 74
3.3.2.4 Configuring the IP Media Settings 75

3.3.2.5 Configuring the General Media Settings 76
3.3.2.6 Configuring the Analog Settings 76
3.3.2.7 Configuring Media Security 77
3.3.3 PSTN Settings 77
3.3.3.1 Configuring the CAS State Machines 77
3.3.3.2 Configuring the Trunk Settings 80
3.3.4 Security Settings 82
3.3.4.1 Configuring the Web User Accounts 83
3.3.4.2 Configuring the Web and Telnet Access List 85
3.3.4.3 Configuring the Firewall Settings 86
3.3.4.4 Configuring the Certificates 89
3.3.4.5 Configuring the General Security Settings 94
3.3.4.6 Configuring the IP Security Proposal Table 95
3.3.4.7 Configuring the IP Security Associations Table 96
3.3.5 Protocol Configuration 99
3.3.5.1 Configuring Media Realms 100
3.3.5.2 Enabling Applications 102
3.3.5.3 Trunk Group 102
3.3.5.4 Protocol Definition 108
3.3.5.5 Application Network Setting 110
3.3.5.6 Proxies, Registration, IP Groups 113
3.3.5.7 Coders and Profile Definitions 127
3.3.5.8 SIP Advanced Parameters 135
3.3.5.9 Manipulation Tables 143
3.3.5.10 Routing Tables 156
3.3.5.11 Endpoint Settings 171
3.3.5.12 Configuring Digital Gateway Parameters 178
3.3.5.13 Configuring the IPmedia Parameters 179
3.3.5.14 SAS Parameters 180
3.3.6 Configuring TDM Bus Settings 184
3.3.7 Advanced Applications 185
3.3.7.1 Configuring Voice Mail Parameters 185
3.3.7.2 Configuring RADIUS Accounting Parameters 186
3.3.7.3 Configuring FXO Parameters 187
3.3.7.4 Configuring LDAP Settings 188
3.4 Management Tab 189
3.4.1 Management Configuration 189
3.4.1.1 Configuring the Management Settings 190
3.4.1.2 Configuring the Regional Settings 196
3.4.1.3 Maintenance Actions 196
3.4.2 Software Update 200
3.4.2.1 Loading Auxiliary Files 200
3.4.2.2 Loading a Software Upgrade Key 202
3.4.2.3 Software Upgrade Wizard 205
3.4.2.4 Backing Up and Restoring Configuration 208
3.5 Status & Diagnostics Tab 209
3.5.1 Status & Diagnostics 209
3.5.1.1 Viewing the Device's Syslog Messages 209
3.5.1.2 Viewing Ethernet Port Information 210
3.5.1.3 Viewing Active IP Interfaces 211
3.5.1.4 Viewing Device Information 212
3.5.1.5 Viewing Performance Statistics 212
3.5.1.6 Viewing Active Alarms 213
3.5.2 Gateway Statistics 214
3.5.2.1 Viewing Call Counters 214
3.5.2.2 Viewing SAS Registered Users 216
3.5.2.3 Viewing Call Routing Status 217
3.5.2.4 Viewing Registration Status 218
3.5.2.5 Viewing IP Connectivity 219

4 INI File Configuration 221

4.1 INI File Format 221
4.1.1 Configuring Individual *ini* File Parameters 222
4.1.2 Configuring *ini* File Table Parameters 222
4.1.3 General *ini* File Formatting Rules 224
4.2 Modifying an *ini* File 225
4.3 Secured Encoded *ini* File 226

5 Element Management System (EMS) 227

5.1 Familiarizing yourself with EMS GUI 227
5.2 Securing EMS-Device Communication 228
5.2.1 Configuring IPSec 228
5.2.2 Changing SSH Login Password 229
5.3 Adding the Device in EMS 230
5.4 Configuring Trunks 232
5.4.1 General Trunk Configuration 232
5.4.2 Configuring ISDN NFAS 233
5.5 Configuring Basic SIP Parameters 236
5.6 Configuring Advanced IPSec/IKE Parameters 237
5.7 Provisioning SIP SRTP Crypto Offered Suites 238
5.8 Provisioning SIP MLPP Parameters 239
5.9 Configuring the Device to Operate with SNMPv3 240
5.9.1 Configuring SNMPv3 using SSH 241
5.9.2 Configuring EMS to Operate with a Pre-configured SNMPv3 System 242
5.9.3 Configuring SNMPv3 to Operate with Non-Configured SNMPv3 System 243
5.9.4 Cloning SNMPv3 Users 244
5.10 Resetting the Device 244
5.11 Upgrading the Device's Software 245

6 Configuration Parameters Reference 247

6.1 Networking Parameters 247
6.1.1 Ethernet Parameters 247
6.1.2 Multiple IP Interfaces and VLAN Parameters 248
6.1.3 Static Routing Parameters 251
6.1.4 Quality of Service Parameters 252
6.1.5 NAT and STUN Parameters 254
6.1.6 NFS Parameters 256
6.1.7 DNS Parameters 257
6.1.8 DHCP Parameters 258
6.1.9 NTP and Daylight Saving Time Parameters 259
6.2 Web and Telnet Parameters 260
6.2.1 General Parameters 260
6.2.2 Web Parameters 261
6.2.3 Telnet Parameters 262
6.3 Debugging and Diagnostics Parameters 263
6.3.1 General Parameters 263
6.3.2 Syslog, CDR and Debug Parameters 265
6.3.3 Remote Alarm Indication Parameters 268
6.3.4 Serial Parameters 269

6.3.5 BootP Parameters 270
6.4 Security Parameters 271
6.4.1 General Parameters 271
6.4.2 HTTPS Parameters 272
6.4.3 SRTP Parameters 274
6.4.4 TLS Parameters 275
6.4.5 SSH Parameters 276
6.4.6 IPSec Parameters 277
6.4.7 OCSP Parameters 279
6.5 RADIUS Parameters 279
6.6 SNMP Parameters 281
6.7 SIP Configuration Parameters 284
6.7.1 General SIP Parameters 284
6.7.2 Network Application Parameters 306
6.7.3 IP Group, Proxy, Registration and Authentication Parameters 308
6.7.4 Voice Mail Parameters 319
6.7.5 Fax and Modem Parameters 323
6.7.6 DTMF and Hook-Flash Parameters 325
6.7.7 Digit Collection and Dial Plan Parameters 330
6.7.8 Coders and Profile Parameters 331
6.8 Supplementary Services Parameters 337
6.8.1 Caller ID Parameters 337
6.8.2 Call Waiting Parameters 342
6.8.3 Call Forwarding Parameters 345
6.8.4 Message Waiting Indication Parameters 347
6.8.5 Call Hold Parameters 349
6.8.6 Call Transfer Parameters 350
6.8.7 Three-Way Conferencing Parameters 351
6.8.8 Emergency Call Parameters 352
6.8.9 FXS Call Cut-Through Parameter 353
6.8.10 Automatic Dialing Parameters 354
6.8.11 Direct Inward Dialing Parameters 355
6.8.12 MLPP Parameters 356
6.9 Standalone Survivability Parameters 359
6.10 IP Media Parameters 363
6.11 PSTN Parameters 375
6.11.1 General Parameters 375
6.11.2 TDM Bus and Clock Timing Parameters 379
6.11.3 CAS Parameters 382
6.11.4 ISDN Parameters 385
6.12 ISDN and CAS Interworking Parameters 392
6.13 Answer and Disconnect Supervision Parameters 409
6.14 Tone Parameters 414
6.14.1 Telephony Tone Parameters 414
6.14.2 Tone Detection Parameters 418
6.14.3 Metering Tone Parameters 419
6.15 Telephone Keypad Sequence Parameters 421
6.16 General FXO Parameters 425
6.17 FXS Parameters 427
6.18 Trunk Groups, Number Manipulation and Routing Parameters 428
6.18.1 Trunk Groups and Routing Parameters 428
6.18.2 Alternative Routing Parameters 435
6.18.3 Number Manipulation Parameters 440

6.18.4 LDAP Parameters 450
6.19 Channel Parameters 451
6.19.1 Voice Parameters 451
6.19.2 Coder Parameters 454
6.19.3 Fax and Modem Parameters 455
6.19.4 DTMF Parameters 461
6.19.5 RTP, RTCP and T.38 Parameters 462
6.20 Auxiliary and Configuration Files Parameters 467
6.20.1 Auxiliary/Configuration File Name Parameters 467
6.20.2 Automatic Update Parameters 468

7 Restoring Factory Default Settings 471

7.1 Restoring Defaults using CLI 471
7.2 Restoring Defaults using an *ini* File 471
7.3 Restoring Defaults using Hardware Reset Button 471

8 Auxiliary Configuration Files 473

8.1 Call Progress Tones File 473
8.1.1 Distinctive Ringing 476
8.1.2 FXS Distinctive Ringing and Call Waiting Tones per Source Number 478
8.2 Prerecorded Tones File 479
8.3 Voice Prompts File 479
8.4 CAS Files 480
8.5 Dial Plan File 480
8.6 User Information File 482

9 IP Telephony Capabilities 485

9.1 Dialing Plan Features 485
9.1.1 Dialing Plan Notation for Routing and Manipulation 485
9.1.2 Digit Mapping 487
9.1.3 External Dial Plan File 488
9.1.3.1 Modifying ISDN-to-IP Calling Party Number 489
9.1.4 Dial Plan Prefix Tags for IP-to-Tel Routing 490
9.2 Routing Applications 492
9.2.1 IP-to-IP Routing Application 492
9.2.1.1 Theory of Operation 493
9.2.1.2 Configuring IP-to-IP Routing 497
9.2.2 Stand-Alone Survivability (SAS) Feature 516
9.2.2.1 Configuring SAS 517
9.2.2.2 Configuring SAS Emergency Calls 518
9.2.3 DSP Channel Resources for SBC/IP-to-IP/IP Media Functionality 519
9.2.4 Multiple SIP Signaling/Media Interfaces Environment 522
9.2.4.1 Media Realms 522
9.2.4.2 Signaling Routing Domain (SRD) Entities 522
9.2.4.3 SIP Interfaces 523
9.2.4.4 Configuration Example 524
9.3 Transcoding using Third-Party Call Control 528
9.3.1 Using RFC 4117 528
9.3.2 Using RFC 4240 (NetAnn Conferencing) 528
9.4 Emergency Phone Number Services - E911 530
9.4.1 FXS Device Emulating PSAP using DID Loop-Start Lines 531

9.4.2 FXO Device Interworking SIP E911 Calls from Service Provider's IP Network to PSAP DID Lines 534
9.5 Routing Based on LDAP Active Directory Queries 538
9.5.1 LDAP Overview 538
9.5.2 AD-Based Tel-to-IP Routing in Microsoft OCS 2007 Environment 539
9.6 Configuring DTMF Transport Types 541
9.7 FXS and FXO Capabilities 542
9.7.1 FXS/FXO Coefficient Types 542
9.7.2 FXO Operating Modes 543
9.7.2.1 FXO Operations for IP-to-Tel Calls 543
9.7.2.2 FXO Operations for Tel-to-IP Calls 546
9.7.2.3 Call Termination on FXO Devices 548
9.7.3 Remote PBX Extension Between FXO and FXS Devices 550
9.7.3.1 Dialing from Remote Extension (Phone at FXS) 551
9.7.3.2 Dialing from PBX Line or PSTN 551
9.7.3.3 Message Waiting Indication for Remote Extensions 551
9.7.3.4 Call Waiting for Remote Extensions 552
9.7.3.5 FXS Gateway Configuration 552
9.7.3.6 FXO Gateway Configuration 553
9.8 Configuring Alternative Routing (Based on Connectivity and QoS) 554
9.8.1 Alternative Routing Mechanism 555
9.8.2 Determining the Availability of Destination IP Addresses 555
9.8.3 PSTN Fallback 555
9.9 Fax and Modem Capabilities 556
9.9.1 Fax/Modem Operating Modes 556
9.9.2 Fax/Modem Transport Modes 556
9.9.2.1 T.38 Fax Relay Mode 557
9.9.2.2 G.711 Fax / Modem Transport Mode 558
9.9.2.3 Fax Fallback 558
9.9.2.4 Fax/Modem Bypass Mode 559
9.9.2.5 Fax / Modem NSE Mode 560
9.9.2.6 Fax / Modem Transparent with Events Mode 561
9.9.2.7 Fax / Modem Transparent Mode 561
9.9.2.8 RFC 2833 ANS Report upon Fax/Modem Detection 562
9.9.3 V.34 Fax Support 562
9.9.3.1 Using Bypass Mechanism for V.34 Fax Transmission 562
9.9.3.2 Using Relay mode for both T.30 and V.34 faxes 563
9.9.4 V.152 Support 563
9.10 Working with Supplementary Services 564
9.10.1 Call Hold and Retrieve 565
9.10.2 Consultation Feature 567
9.10.3 Call Transfer 568
9.10.4 Call Forward 568
9.10.4.1 Call Forward Reminder Ring 570
9.10.4.2 Call Forward Reminder (Off-Hook) Special Dial Tone 570
9.10.5 Call Waiting 571
9.10.6 Message Waiting Indication 572
9.10.7 Caller ID 574
9.10.7.1 Caller ID Detection / Generation on the Tel Side 574
9.10.7.2 Debugging a Caller ID Detection on FXO 575
9.10.7.3 Caller ID on the IP Side 576
9.10.8 Three-Way Conferencing 577
9.11 Routing Examples 578
9.11.1 SIP Call Flow Example 578
9.11.2 SIP Authentication Example 580
9.11.3 Proxy or Registrar Registration Example 583
9.11.4 Establishing a Call between Two Devices 584

9.11.5 Trunk-to-Trunk Routing Example 585
9.11.6 SIP Trunking between Enterprise and ITSPs 586
9.12 Mapping PSTN Release Cause to SIP Response 589
9.13 Querying Device Channel Resources using SIP OPTIONS 589
9.14 Answer Machine Detector (AMD) 590
9.15 Event Notification using X-Detect Header 594
9.16 Supported RADIUS Attributes 596
9.17 Call Detail Record 599
9.18 RTP Multiplexing (ThroughPacket) 601
9.19 Dynamic Jitter Buffer Operation 602

10 Networking Capabilities 603

10.1 Ethernet Interface Configuration 603
10.2 Ethernet Interface Redundancy 604
10.3 NAT (Network Address Translation) Support 604
10.3.1 STUN 605
10.3.2 First Incoming Packet Mechanism 606
10.3.3 No-Op Packets 606
10.4 IP Multicasting 606
10.5 Robust Receipt of RTP Streams 607
10.6 Multiple Routers Support 607
10.7 Simple Network Time Protocol Support 607
10.8 IP QoS via Differentiated Services (DiffServ) 608
10.9 Network Configuration 608
10.9.1 Multiple Network Interfaces and VLANs 609
10.9.1.1 Overview of Multiple Interface Table 610
10.9.1.2 Columns of the Multiple Interface Table 610
10.9.1.3 Other Related Parameters 613
10.9.1.4 Multiple Interface Table Configuration Summary and Guidelines 616
10.9.1.5 Troubleshooting the Multiple Interface Table 617
10.9.2 Routing Table 618
10.9.2.1 Routing Table Overview 618
10.9.2.2 Routing Table Columns 618
10.9.2.3 Routing Table Configuration Summary and Guidelines 620
10.9.2.4 Troubleshooting the Routing Table 620
10.9.3 Setting up the Device 621
10.9.3.1 Using the Web Interface 621
10.9.3.2 Using the *ini* File 621

11 Advanced PSTN Configuration 625

11.1 Clock Settings 625
11.2 Release Reason Mapping 626
11.2.1 Reason Header 626
11.2.2 Fixed Mapping of ISDN Release Reason to SIP Response 626
11.2.3 Fixed Mapping of SIP Response to ISDN Release Reason 628
11.3 ISDN Overlap Dialing 630
11.4 ISDN Non-Facility Associated Signaling (NFAS) 631
11.4.1 NFAS Interface ID 632
11.4.2 Working with DMS-100 Switches 632

11.4.3 Creating an NFAS-Related Trunk Configuration 633
11.5 Redirect Number and Calling Name (Display) 634
11.6 Automatic Gain Control (AGC) 634
12 Tunneling Applications 635
12.1 TDM Tunneling 635
12.1.1 DSP Pattern Detector 638
12.2 QSIG Tunneling 638
13 IP Media Capabilities 641
13.1 Conference Server 641
13.1.1 Simple Conferencing (NetAnn) 642
13.1.1.1 SIP Call Flow 642
13.1.1.2 Creating a Conference 643
13.1.1.3 Joining a Conference 643
13.1.1.4 Terminating a Conference 643
13.1.1.5 PSTN Participants 644
13.1.2 Advanced Conferencing (MSCML) 644
13.1.2.1 Creating a Conference 644
13.1.2.2 Joining a Conference 645
13.1.2.3 Modifying a Conference 646
13.1.2.4 Applying Media Services on a Conference 648
13.1.2.5 Active Speaker Notification 648
13.1.2.6 Terminating a Conference 649
13.1.3 Conference Call Flow Example 650
13.2 Announcement Server 656
13.2.1 NetAnn Interface 656
13.2.1.1 Playing a Local Voice Prompt 656
13.2.1.2 Playing using HTTP/NFS Streaming 656
13.2.1.3 Supported Attributes 657
13.2.2 MSCML Interface 658
13.2.2.1 Operation 659
13.2.2.2 Operating with Audio Bundles 660
13.2.2.3 Playing Announcements 662
13.2.2.4 Playing Announcements and Collecting Digits 662
13.2.2.5 Playing Announcements and Recording Voice 664
13.2.2.6 Stopping the Playing of an Announcement 665
13.2.2.7 Relevant Parameters 665
13.2.2.8 Signal Events Notifications 666
13.2.3 Voice Streaming 667
13.2.3.1 Voice Streaming Features 667
13.2.3.2 Using File Coders with Different Channel Coders 670
13.2.3.3 Maximum Concurrent Playing and Recording 671
13.2.3.4 LBR Coders Support 671
13.2.3.5 HTTP Recording Configuration 672
13.2.3.6 NFS Configuration Using the *ini* File 672
13.2.3.7 Supported HTTP Servers 673
13.2.3.8 Supporting NFS Servers 674
13.2.3.9 Common Troubleshooting 677
13.2.4 Announcement Call Flow Example 678
13.3 Voice XML Interpreter 681
13.3.1 Features 681
13.3.2 Feature Key 681
13.3.3 VXML Scripts 682
13.3.4 Proprietary Extensions 682
13.3.4.1 Record 683

13.3.4.2 Audio Extensions 684
13.3.4.3 Language Identifier Support 687
13.3.5 Combining <audio> Elements 688
13.3.6 Notes Regarding Non-compliant Functionality 688
13.3.7 Supported Elements and Attributes 688
13.3.7.1 VoiceXML Supported Elements and Attributes 689
13.3.7.2 SRGS and SSML Support 698
13.3.7.3 VoiceXML Supported Properties 699
13.3.7.4 VoiceXML Variables and Events 700
13.3.7.5 ECMAScript Support 702
13.3.8 Example of UDT 'beep' Tone Definition 703
13.3.9 Limitations and Restrictions 703

14 SIP Software Package 705

15 Selected Technical Specifications 707

15.1 Mediant 1000 707
15.2 Mediant 600 709

List of Figures

Figure 3-1: Enter Network Password Screen 29
Figure 3-2: Main Areas of the Web Interface GUI 30
Figure 3-3: "Reset" Displayed on Toolbar 31
Figure 3-4: Terminology for Navigation Tree Levels 32
Figure 3-5: Navigation Tree in Basic and Full View 33
Figure 3-6: Showing and Hiding Navigation Pane 34
Figure 3-7: Toggling between Basic and Advanced Page View 36
Figure 3-8: Expanding and Collapsing Parameter Groups 37
Figure 3-9: Editing Symbol after Modifying Parameter Value 38
Figure 3-10: Value Reverts to Previous Valid Value 39
Figure 3-11: Adding an Index Entry to a Table 40
Figure 3-12: Compacting a Web Interface Table 41
Figure 3-13: Searched Result Screen 42
Figure 3-14: Scenario Creation Confirm Message Box 43
Figure 3-15: Creating a Scenario 44
Figure 3-16: Scenario Loading Message Box 45
Figure 3-17: Scenario Example 45
Figure 3-18: Scenario File Page 48
Figure 3-19: Scenario Loading Message Box 49
Figure 3-20: Message Box for Confirming Scenario Deletion 49
Figure 3-21: Confirmation Message Box for Exiting Scenario Mode 50
Figure 3-22: User-Defined Web Welcome Message after Login 51
Figure 3-23: Help Topic for Current Page 52
Figure 3-24: Log Off Confirmation Box 53
Figure 3-25: Web Session Logged Off 53
Figure 3-26: Mediant 600 Home Page 54
Figure 3-27: Mediant 1000 Home Page 54
Figure 3-28: Shortcut Menu (Example, Mediant 1000) 56
Figure 3-29: Text Box (Example, Mediant 1000) 56
Figure 3-30: Shortcut Menu (Example, Mediant 1000) 57
Figure 3-31: Selecting Port Settings from Shortcut Menu 57
Figure 3-32: Basic Information Screen 57
Figure 3-33: Trunks and Channels Status Screen 58
Figure 3-34: Basic Channel Information Page 58
Figure 3-35: Remove Module Button Appears after Clicking Module Name 59
Figure 3-36: Module Removal Confirmation Message Box 59
Figure 3-37: Removed Module 60
Figure 3-38: Insert Module Button after Clicking Module's Name 60
Figure 3-39: IP Settings Page 62
Figure 3-40: Confirmation Message for Accessing the Multiple Interface Table 62
Figure 3-41: Multiple Interface Table Page 62
Figure 3-42: Application Settings Page 66
Figure 3-43: NFS Settings Page 67
Figure 3-44: IP Routing Table Page 69
Figure 3-45: QoS Settings Page 70
Figure 3-46: Voice Settings Page 72
Figure 3-47: Fax/Modem/CID Settings Page 73
Figure 3-48: RTP/RTCP Settings Page 74
Figure 3-49: IPMedia Settings Page 75
Figure 3-50: General Media Page 76
Figure 3-51: Analog Settings Page 76
Figure 3-52: Media Security Page 77
Figure 3-53: CAS State Machine Page 78
Figure 3-54: Trunk Settings Page 80
Figure 3-55: Trunk Scroll Bar 81
Figure 3-56: WEB User Accounts Page (for Users with 'Security Administrator' Privileges) 84
Figure 3-57: Web & Telnet Access List Page - Add New Entry 85

Figure 3-58: Web & Telnet Access List Table 86
Figure 3-59: Firewall Settings Page 87
Figure 3-60: Certificates Signing Request Page 90
Figure 3-61: IKE Table Listing Loaded Certificate Files 91
Figure 3-62: General Security Settings Page 94
Figure 3-63: IP Security Proposals Table 95
Figure 3-64: IP Security Associations Table Page 96
Figure 3-65: SIP Media Realm Table Page 100
Figure 3-66: Applications Enabling Page 102
Figure 3-67: Trunk Group Table Page 103
Figure 3-68: Trunk Group Settings Page 105
Figure 3-69: SIP General Parameters Page 109
Figure 3-70: DTMF & Dialing Page 110
Figure 3-71: SRD Table Page 111
Figure 3-72: SRD Table Page 112
Figure 3-73: IP Group Page 114
Figure 3-74: Account Table Page 119
Figure 3-75: Proxy & Registration Page 122
Figure 3-76: Proxy Sets Table Page 124
Figure 3-77: Coders Page 129
Figure 3-78: Coder Group Settings Page 130
Figure 3-79: Tel Profile Settings Page 132
Figure 3-80: IP Profile Settings Page 134
Figure 3-81: Advanced Parameters Page 136
Figure 3-82: Supplementary Services Page 138
Figure 3-83: Metering Tones Page 139
Figure 3-84: Charge Codes Table Page 140
Figure 3-85: Keypad Features Page 142
Figure 3-86: General Settings Page 143
Figure 3-87: Source Phone Number Manipulation Table for Tel-to-IP Calls 146
Figure 3-88: Reditect Number IP to Tel Page 149
Figure 3-89: Redirect Number Tel to IP Page 151
Figure 3-90: Phone Context Table Page 153
Figure 3-91: Reasons for Alternative Routing Page 157
Figure 3-92: Routing General Parameters Page 158
Figure 3-93: Outbound IP Routing Table Page 160
Figure 3-94: Inbound IP Routing Table 164
Figure 3-95: Internal DNS Table Page 167
Figure 3-96: Internal SRV Table Page 168
Figure 3-97: Release Cause Mapping Page 169
Figure 3-98: Forward on Busy Trunk Destination Page 170
Figure 3-99: Authentication Page 172
Figure 3-100: Automatic Dialing Page 173
Figure 3-101: Caller Display Information Page 174
Figure 3-102: Call Forward Table Page 175
Figure 3-103: Caller ID Permissions Page 176
Figure 3-104: Caller Waiting Page 177
Figure 3-105: Digital Gateway Parameters Page 178
Figure 3-106: IP Media Settings Page 179
Figure 3-107: SAS Configuration Page 181
Figure 3-108: IP2IP Routing Page 182
Figure 3-109: TDM Bus Settings Page 184
Figure 3-110: Voice Mail Settings Page 186
Figure 3-111: RADIUS Parameters Page 187
Figure 3-112: FXO Settings Page 187
Figure 3-113: LDAP Settings Page 188
Figure 3-114: Management Settings Page 190
Figure 3-115: SNMP Trap Destinations Page 191

Figure 3-116: SNMP Community Strings Page 192
Figure 3-117: SNMP V3 Setting Page 193
Figure 3-118: SNMP Trusted Managers 195
Figure 3-119: Regional Settings Page 196
Figure 3-120: Maintenance Actions Page 197
Figure 3-121: Reset Confirmation Message Box 198
Figure 3-122: Device Lock Confirmation Message Box 199
Figure 3-123: Load Auxiliary Files Page 201
Figure 3-124: Software Upgrade Key Status Page 203
Figure 3-125: Software Upgrade Key with Multiple S/N Lines 204
Figure 3-126: Start Software Upgrade Wizard Screen 206
Figure 3-127: End Process Wizard Page 207
Figure 3-128: Configuration File Page 208
Figure 3-129: Message Log Screen 210
Figure 3-130: Ethernet Port Information Screen 211
Figure 3-131: IP Interface Status Page 211
Figure 3-132: Device Information Page 212
Figure 3-133: Performance Statistics Page 213
Figure 3-134: Calls Count Page 214
Figure 3-135: SAS Registered Users Page 216
Figure 3-136: Call Routing Status Page 217
Figure 3-137: Registration Status Page 218
Figure 3-138: IP Connectivity Page 219
Figure 5-1: Areas of the EMS GUI 227
Figure 5-2: EMS Login Screen 230
Figure 5-3: Adding a Region 231
Figure 5-4: Defining the IP Address 231
Figure 5-5: DS Trunks List 232
Figure 5-6: Trunks Channels Table 232
Figure 5-7: General Settings Screen 233
Figure 5-8: EMS ISDN Settings Screen 234
Figure 5-9: SIP Protocol Definitions Frame 236
Figure 5-10: IPSec Table Screen 238
Figure 5-11: Authentication & Security Screen 239
Figure 5-12: MLPP Screen 240
Figure 5-13: MG Information Screen 242
Figure 5-14: SNMP Configuration Screen 243
Figure 5-15: Confirmation for Saving Configuration and Resetting Device 244
Figure 5-16: Software Manager Screen 245
Figure 5-17: Add Files Screen 245
Figure 5-18: Files Manager Screen 246
Figure 8-1: Example of a User Information File 483
Figure 9-1: Prefix to Add Field with Notation 486
Figure 9-2: Configuring Dial Plan File Label for IP-to-Tel Routing 491
Figure 9-3: Configuring Manipulation for Removing Label 491
Figure 9-4: Basic Schema of the Device's IP-to-IP Call Handling 493
Figure 9-5: IP-to-IP Routing/Registration/Authentication of Remote IP-PBX Users (Example) 495
Figure 9-6: IP-to-IP Routing for IP-PBX Remote Users in Survivability Mode (Example) 496
Figure 9-7: Registration with Multiple ITSP's on Behalf of IP-PBX 497
Figure 9-8: SIP Trunking Setup Scenario Example 499
Figure 9-9: Defining Required Media Channels 500
Figure 9-10: Proxy Set ID #1 for ITSP-A 502
Figure 9-11: Proxy Set ID #2 for ITSP-B 503
Figure 9-12: Proxy Set ID #3 for the IP-PBX 504
Figure 9-13: Defining IP Group 1 505
Figure 9-14: Defining IP Group 2 506
Figure 9-15: Defining IP Group 3 507
Figure 9-16: Defining IP Group 4 508
Figure 9-17: Defining Accounts for Registration 509

Figure 9-18: Defining Coder Group ID 1 510
Figure 9-19: Defining Coder Group ID 2 510
Figure 9-20: Defining IP Profile ID 1 511
Figure 9-21: Defining Inbound IP Routing Rules 512
Figure 9-22: Defining Outbound IP Routing Rules 513
Figure 9-23: Defining Destination Phone Number Manipulation Rules 515
Figure 9-24: SAS Routing in Emergency Mode 517
Figure 9-25: SAS Routing for Emergency Calls 518
Figure 9-26: Multiple SIP Signaling/RTP Interfaces Scenario 523
Figure 9-27: Multiple SIP Signaling/RTP Interfaces Example 524
Figure 9-28: Defining Trunk Group 525
Figure 9-29: Defining IP Interfaces 525
Figure 9-30: Defining Media Realms 525
Figure 9-31: Defining SRDs 525
Figure 9-32: Defining SIP Interfaces 526
Figure 9-33: Defining Proxy Set 526
Figure 9-34: Defining IP Groups 527
Figure 9-35: Defining IP-to-Trunk Group Routing 527
Figure 9-36: Defining Trunk Group to IP Group Routing 527
Figure 9-37: Direct Connection (Example) 529
Figure 9-38: FXS Device Emulating PSAP using DID Loop-Start Lines 531
Figure 9-39: FXO Device Interfacing between E911 Switch and PSAP 534
Figure 9-40: Active Directory-based Routing Rules in Outbound IP Routing Table 540
Figure 9-41: Call Flow for One-Stage Dialing 544
Figure 9-42: Call Flow for Two-Stage Dialing 545
Figure 9-43: Call Flow for Automatic Dialing 546
Figure 9-44: Call Flow for Collecting Digits Mode 547
Figure 9-45: FXO-FXS Remote PBX Extension (Example) 550
Figure 9-46: MWI for Remote Extensions 552
Figure 9-47: Call Waiting for Remote Extensions 552
Figure 9-48: Assigning Phone Numbers to FXS Endpoints 552
Figure 9-49: Automatic Dialing for FXS Ports 553
Figure 9-50: FXS Tel-to-IP Routing Configuration 553
Figure 9-51: Assigning Phone Numbers to FXO Ports 553
Figure 9-52: FXO Automatic Dialing Configuration 554
Figure 9-53: FXO Tel-to-IP Routing Configuration 554
Figure 9-54: Double Hold SIP Call Flow 566
Figure 9-55: Call Forward Reminder with Application Server 570
Figure 9-56: SIP Call Flow 578
Figure 9-57: Assigning Phone Numbers to Device 10.2.37.10 584
Figure 9-58: Assigning Phone Numbers to Device 10.2.37.20 584
Figure 9-59: Routing Calls Between Devices 585
Figure 9-60: Configuring Proxy Set ID #1 in the Proxy Sets Table Page 586
Figure 9-61: Configuring Proxy Set ID #1 in the Proxy Sets Table Page 587
Figure 9-62: Configuring IP Groups #1 and #2 in the IP Group Table Page 587
Figure 9-63: Assigning Trunks to Trunk Group ID #1 588
Figure 9-64: Configuring Trunk Group #1 for Registration per Account in Trunk Group Settings Page 588
Figure 9-65: Configuring Accounts for PBX Registration to ITSPs in Account Table Page 588
Figure 9-66: Configuring ITSP-to-Trunk Group #1 Routing in IP to Trunk Group Table Page 588
Figure 9-67: Configuring Tel-to-IP Routing to ITSPs in Tel to IP Routing Table Page 588
Figure 10-1: NAT Support 604
Figure 10-2: Multiple Network Interfaces 609
Figure 10-3: Prefix Length and Subnet Masks Columns 619
Figure 10-4: Interface Column 619
Figure 13-1: Announcement Call Flow 678

List of Tables

Table 3-1: Description of Toolbar Buttons 31
Table 3-2: *ini* File Parameter for Welcome Login Message 51
Table 3-3: Description of the Areas of the Home Page 54
Table 3-4: Color-Coding Status for Trunk's Channels 58
Table 3-5: Multiple Interface Table Parameters Description 63
Table 3-6: NFS Settings Parameters 68
Table 3-7: IP Routing Table Description 69
Table 3-8: CAS State Machine Parameters Description 79
Table 3-9: Web User Accounts Access Levels and Privileges 83
Table 3-10: Default Attributes for the Web User Accounts 83
Table 3-11: Internal Firewall Parameters 88
Table 3-12: IP Security Proposals Table Configuration Parameters 95
Table 3-13: Default IPSec/IKE Proposals 96
Table 3-14: IP Security Associations Table Configuration Parameters 97
Table 3-15: SIP Media Realm Table Parameters 100
Table 3-16: Trunk Group Table Parameters 103
Table 3-17: Trunk Group Settings Parameters 106
Table 3-18: SRD Table Parameters 111
Table 3-19: SIP Interface Table Parameters 112
Table 3-20: IP Group Parameters 115
Table 3-21: Account Table Parameters Description 119
Table 3-22: Proxy Sets Table Parameters 124
Table 3-23: Description of Parameter Unique to IP Profile 135
Table 3-24: Number Manipulation Parameters Description 146
Table 3-25: Redirect Number IP to Tel Parameters Description 149
Table 3-26: Redirect Number Tel to IP Parameters Description 152
Table 3-27: Phone-Context Parameters Description 154
Table 3-28: NPI/TON Values for ISDN ETSI 155
Table 3-29: Outbound IP Routing Table Parameters 161
Table 3-30: inbound IP Routing Table Description 165
Table 3-31: Call Forward Table 175
Table 3-32: SAS Routing Table Parameters 182
Table 3-33: SNMP Trap Destinations Parameters Description 191
Table 3-34: SNMP Community Strings Parameters Description 193
Table 3-35: SNMP V3 Users Parameters 194
Table 3-36: Auxiliary Files Descriptions 200
Table 3-37: Ethernet Port Information Parameters 211
Table 3-38: Call Counters Description 215
Table 3-39: SAS Registered Users Parameters 216
Table 3-40: Call Routing Status Parameters 217
Table 3-41: IP Connectivity Parameters 219
Table 6-1: Ethernet Parameters 247
Table 6-2: IP Network Interfaces and VLAN Parameters 248
Table 6-3: Static Routing Parameters 251
Table 6-4: QoS Parameters 252
Table 6-5: NAT and STUN Parameters 254
Table 6-6: NFS Parameters 256
Table 6-7: DNS Parameters 257
Table 6-8: DHCP Parameters 258
Table 6-9: NTP and Daylight Saving Time Parameters 259
Table 6-10: General Web and Telnet Parameters 260
Table 6-11: Web Parameters 261
Table 6-12: Telnet Parameters 262
Table 6-13: General Debugging and Diagnostic Parameters 263
Table 6-14: Syslog, CDR and Debug Parameters 265
Table 6-15: RAI Parameters 268
Table 6-16: Serial Parameters 269

Table 6-17: BootP Parameters 270
Table 6-18: General Security Parameters 271
Table 6-19: HTTPS Parameters 272
Table 6-20: SRTP Parameters 274
Table 6-21: TLS Parameters 275
Table 6-22: SSH Parameters 276
Table 6-23: IPSec Parameters 277
Table 6-24: OCSP Parameters 279
Table 6-25: RADIUS Parameters 279
Table 6-26: SNMP Parameters 281
Table 6-27: General SIP Parameters 284
Table 6-28: SIP Network Application Parameters 306
Table 6-29: Proxy, Registration and Authentication SIP Parameters 308
Table 6-30: Voice Mail Parameters 319
Table 6-31: Fax and Modem Parameters 323
Table 6-32: DTMF and Hook-Flash Parameters 325
Table 6-33: Digit Collection and Dial Plan Parameters 330
Table 6-34: Profile Parameters 331
Table 6-35: Caller ID Parameters 337
Table 6-36: Call Waiting Parameters 342
Table 6-37: Call Forwarding Parameters 345
Table 6-38: MWI Parameters 347
Table 6-39: Call Hold Parameters 349
Table 6-40: Call Transfer Parameters 350
Table 6-41: Three-Way Conferencing Parameters 351
Table 6-42: Emergency Call Parameters 352
Table 6-43: Call Cut-Through Parameter 353
Table 6-44: Automatic Dialing Parameters 354
Table 6-45: DID Parameters 355
Table 6-46: MLPP Parameters 356
Table 6-47: SAS Parameters 359
Table 6-48: IP Media Parameters 363
Table 6-49: General PSTN Parameters 375
Table 6-50: TDM Bus and Clock Timing Parameters 379
Table 6-51: CAS Parameters 382
Table 6-52: ISDN Parameters 385
Table 6-53: ISDN and CAS Interworking Parameters 392
Table 6-54: Answer and Disconnect Parameters 409
Table 6-55: Tone Parameters 414
Table 6-56: Tone Detection Parameters 418
Table 6-57: Metering Tone Parameters 419
Table 6-58: Keypad Sequence Parameters 421
Table 6-59: General FXO Parameters 425
Table 6-60: General FXS Parameters 427
Table 6-61: Routing Parameters 428
Table 6-62: Alternative Routing Parameters 435
Table 6-63: Number Manipulation Parameters 440
Table 6-64: LDAP Parameters 450
Table 6-65: Voice Parameters 451
Table 6-66: Coder Parameters 454
Table 6-67: Fax and Modem Parameters 455
Table 6-68: DTMF Parameters 461
Table 6-69: RTP/RTCP and T.38 Parameters 462
Table 6-70: Auxiliary and Configuration File Parameters 467
Table 6-71: Automatic Update of Software and Configuration Files Parameters 468
Table 8-1: User Information Items 482
Table 9-1: Dialing Plan Notations 485
Table 9-2: Digit Map Pattern Notations 487

Table 9-3: Dialed MF Digits Sent to PSAP 533
Table 9-4: Dialed Number by Device Depending on Calling Number 535
Table 9-5: Approximate AMD Detection Normal Sensitivity (Based on North American English) 590
Table 9-6: Approximate AMD Detection High Sensitivity (Based on North American English)..... 591
Table 9-7: Supported X-Detect Event Types 594
Table 9-8: Special Information Tones (SITs) Reported by the device..... 595
Table 9-9: Supported RADIUS Attributes 596
Table 9-10: Supported CDR Fields..... 599
Table 10-1: Multiple Interface Table 610
Table 10-2: Application Types 611
Table 10-3: Configured Default Gateway Example 612
Table 10-4: Separate Routing Table Example 612
Table 10-5: Quality of Service Parameters..... 614
Table 10-6: Traffic / Network Types and Priority 615
Table 10-7: Application Type Parameters 616
Table 10-8: Routing Table Layout 618
Table 10-9: Multiple Interface Table - Example1 622
Table 10-10: Routing Table - Example 1 622
Table 10-11: Multiple Interface Table - Example 2 623
Table 10-12: Routing Table - Example 2 623
Table 10-13: Multiple Interface Table - Example 3 624
Table 10-14: Routing Table - Example 3 624
Table 11-1: Mapping of ISDN Release Reason to SIP Response 626
Table 11-2: Mapping of SIP Response to ISDN Release Reason 628
Table 11-3: Calling Name (Display)..... 634
Table 11-4: Redirect Number 634
Table 13-1: Simple Conferencing SIP Call Flow 642
Table 13-2: Advanced Conferencing SIP Call Flow 645
Table 13-3: MSCML Conferencing with Personalized Mixes 646
Table 13-4: Modifying a Conference - SIP Call Flow..... 647
Table 13-5: Applying Media Services on a Conference - SIP Call Flow 648
Table 13-6: Terminating a Conference - SIP Call Flow 649
Table 13-7: Conference Call Flow Example 650
Table 13-8: MSCML Architecture 658
Table 13-9: Reportable Events 666
Table 13-10: Coder Combinations - Playing a File..... 670
Table 13-11: Coder Combinations - Recording a File 671
Table 13-12: DSP Templates 671
Table 13-13: Compatible NFS Servers..... 674
Table 13-14: Troubleshooting..... 677
Table 13-15: Say-as Phrase Types 684
Table 13-16: Support for Language Code Portion of Identifier..... 687
Table 13-17: VoiceXML Supported Elements and Attributes 689
Table 13-18: VoiceXML Supported Properties 699
Table 13-19: VoiceXML Variables and Events 700
Table 13-20: ECMAScript Support 702
Table 14-1: Software Package 705
Table 15-1: Mediant 1000 Functional Specifications..... 707
Table 15-2: Mediant 600 Functional Specifications..... 709

Notice

This document describes the AudioCodes Mediant 1000 and Mediant 600 Voice-over-IP (VoIP) SIP media gateways.

Information contained in this document is believed to be accurate and reliable at the time of printing. However, due to ongoing product improvements and revisions, AudioCodes cannot guarantee accuracy of printed material after the Date Published nor can it accept responsibility for errors or omissions. Before consulting this document, check the corresponding Release Notes regarding feature preconditions and/or specific support in this release. In cases where there are discrepancies between this document and the Release Notes, the information in the Release Notes supersedes that in this document. Updates to this document and other documents can be viewed by registered customers at http://www.audiocodes.com/downloads.

© Copyright 2010 AudioCodes Ltd. All rights reserved.

This document is subject to change without notice.

Date Published: March-14-2010

Trademarks

AudioCodes, AC, AudioCoded, Ardito, CTI2, CTI², CTI Squared, HD VoIP, HD VoIP Sounds Better, InTouch, IPmedia, Mediant, MediaPack, NetCoder, Netrake, Nuera, Open Solutions Network, OSN, Stretto, TrunkPack, VMAS, VoicePacketizer, VoIPerfect, VoIPerfectHD, What's Inside Matters, Your Gateway To VoIP and 3GX are trademarks or registered trademarks of AudioCodes Limited. All other products or trademarks are property of their respective owners. Product specifications are subject to change without notice.

WEEE EU Directive

Pursuant to the WEEE EU Directive, electronic and electrical waste must not be disposed of with unsorted waste. Please contact your local recycling authority for disposal of this product.

Customer Support

Customer technical support and service are provided by AudioCodes' Distributors, Partners, and Resellers from whom the product was purchased. For Customer support for products purchased directly from AudioCodes, contact support@audiocodes.com.

Abbreviations and Terminology

Each abbreviation, unless widely used, is spelled out in full when first used. Only industry-standard terms are used throughout this manual. Hexadecimal notation is indicated by 0x preceding the number.

Related Documentation

Manual Name
Product Reference Manual SIP CPE Devices
Mediant 1000 & Mediant 600 SIP Release Notes
Mediant 1000 Installation Manual
Mediant 600 Installation Manual
CPE Configuration Guide for IP Voice Mail

Note: The term *device* refers to the Mediant 1000 and Mediant 600 gateways.

Note: Before configuring the device, ensure that it is installed correctly as instructed in the device's *Installation Manual*.

Note: For assigning an IP address to the device, refer to the device's *Installation Manual*.

Note: The terms *IP-to-Tel* and *Tel-to-IP* refer to the direction of the call relative to the device. *IP-to-Tel* refers to calls received from the IP network and destined to the PSTN/PBX (i.e., telephone connected directly or indirectly to the device); *Tel-to-IP* refers to calls received from the PSTN/PBX and destined for the IP network.

Notes:

- FXO (Foreign Exchange Office) is the interface replacing the analog telephone and connects to a Public Switched Telephone Network (PSTN) line from the Central Office (CO) or to a Private Branch Exchange (PBX). The FXO is designed to receive line voltage and ringing current, supplied from the CO or the PBX (just like an analog telephone). An FXO VoIP device interfaces between the CO/PBX line and the Internet.
- FXS (Foreign Exchange Station) is the interface replacing the Exchange (i.e., the CO or the PBX) and connects to analog telephones, dial-up modems, and fax machines. The FXS is designed to supply line voltage and ringing current to these telephone devices. An FXS VoIP device interfaces between the analog telephone devices and the Internet.

1 Overview

This section provides an overview of the Mediant 1000 and Mediant 600 media gateways.

1.1 Mediant 1000

The Mediant 1000 (hereafter referred to as *device*) is a best-of-breed Voice-over-IP (VoIP) Session Initiation Protocol (SIP) Media Gateway, using field-proven, market-leading technology, implementing analog and digital cutting-edge technology. The device is designed to seamlessly interface between Time-Division Multiplexing (TDM) and Internet Protocol (IP) networks, providing superior voice quality and optimized packet voice streaming (voice, fax, and data traffic) over IP networks.

The device is best suited for small-to-medium sized (SME) enterprises, branch offices, and residential media gateway solutions. The device is a highly scalable and modular system that matches the density requirements for smaller environments, while meeting service providers' demands for growth.

The device is ideal for connecting an enterprise's legacy telephones, fax machines, and Private Branch Exchange (PBX) systems to IP-based telephony networks, as well as for seamlessly connecting IP-based PBX architecture to the Public Switched Telephone Network (PSTN). The device also provides SIP trunking capabilities (including IP-to-IP call routing) for Enterprises operating with multiple Internet Telephony Service Providers (ITSP) for VoIP services. In addition to operating as a pure media gateway, the device incorporates an open platform, known as the Open Solutions Network (OSN) server, allowing additional deployment options by hosting third-party partner VoIP applications such as IP-PBX, Calling Card, and IP-PBX redundancy.

The device also provides conferencing services over VoIP networks. This is supported by an optional Media Processing Module (MPM) that can be housed in the device's chassis. The MPM module also provides IP Media channels for use on various Media Server applications.

The device is fully interoperable with multiple vendor gateways, softswitches, SIP servers, gatekeepers, proxy servers, IP phones, session border controllers (SBC), and firewalls. The device is designed to meet regulatory approval (including Safety, EMC, and Telecom for USA, EU and other countries).

Intelligently packaged in a stackable and compact 1U chassis, it can be mounted on a desk, a wall, or in a standard 19-inch rack. The device is supplied with two integral mounting brackets for facilitating rack installation.

The device is equipped with two 10/100Base-TX Ethernet ports for connection to the IP network. The second Ethernet port is used for 1+1 Ethernet redundancy.

The device supports mixed digital and analog interface configurations:

- **Digital:**
  - The device supports multiples of 1, 2, or 4 E1/T1/J1 spans for connecting the PSTN/PBX to the IP network. The digital modules provide RJ-48 ports. The digital module can be configured with up to 1 or 2 paired spans for switching to the PSTN in case of power or network failure (PSTN Fallback).
  - The device also supports ISDN Basic Rate Interface (BRI) modules for connecting BRI-based PSTN or PBX lines to the IP network. Each BRI module supports four BRI ports (RJ-45). Up to five BRI modules can be housed in the device, supporting up to 20 BRI digital ports. The BRI module can be configured as 'Lifeline' telephone interfaces, switching to the PSTN in case of power failure or network problems.

- Depending on configuration, the device can provide IP Media channels at the expense of PSTN channels. These channels may be used for Media Server applications.

- **Analog:** The device's analog interface supports up to 24 analog ports (four ports per analog module) in various Foreign Exchange Office (FXO) or Foreign Exchange Station (FXS) configurations, supporting up to 24 simultaneous VoIP calls. The device supports up to six analog modules, each module providing four analog RJ-11 ports. The FXO module can be used to connect analog lines of an enterprise's PBX or the PSTN to the IP network. The FXS module can be used to connect legacy telephones, fax machines, and modems to the IP network. Optionally, the FXS module can be connected to the external trunk lines of a PBX. When deployed with a combination of FXO and FXS modules, the device can be used as a PBX for Small Office Home Office (SOHO) users, and businesses not equipped with a PBX.

- **Media Processing Module (MPM):** The MPM module provides IP media channels for conferencing and media server functionality. The device can house up to three MPM modules.

The device has enhanced hardware and software capabilities to ease its installation and to maintain voice quality. If the measured voice quality falls beneath a pre-configured value, or the path to the destination is disconnected, the device assures voice connectivity by 'falling' back to the PSTN. In the event of network problems or power failures, calls can be routed back to the PSTN without requiring routing modifications in the PBX. Further reliability is provided by dual Ethernet ports and an optional dual AC power supply.

The device supports various ISDN PRI protocols such as EuroISDN, North American NI2, Lucent™ 4/5ESS, Nortel™ DMS100 and others. It also supports various ISDN BRI protocols such as ETSI 5ESS and QSIG over BRI. In addition, it supports different variants of CAS protocols for E1 and T1 spans, including MFC R2, E&M immediate start, E&M delay dial / start, loop start and ground start.

The device provides a variety of management and provisioning tools, including an HTTP-based embedded Web server, Telnet, Element Management System (EMS), and Simple Network Management Protocol (SNMP). The user-friendly, Web interface provides remote configuration using a Web browser (such as Microsoft™ Internet Explorer™).

1.2 Mediant 600

The Mediant 600 (hereafter referred to as *device*) is a cost-effective, wireline Voice-over-IP (VoIP) Session Initiation Protocol (SIP)-based media gateway. It is designed to interface between Time-Division Multiplexing (TDM) and IP networks in enterprises, small and medium businesses (SMB), and CPE application service providers. Incorporating AudioCodes' innovative VoIP technology, the device enables rapid time-to-market and reliable cost-effective deployment of next-generation networks.

The device is based on VoIPerfect, AudioCodes underlying, best-of-breed, media gateway core technology. The device provides superior voice technology for connecting legacy telephone and PBX systems to IP networks, as well as seamlessly connecting IP-PBXs to the PSTN. The device also provides SIP trunking capabilities for Enterprises operating with multiple Internet Telephony Service Providers (ITSP) for VoIP services. The device is fully interoperable with multiple vendors of IP-PBXs, IP Centrex application servers, softswitches, gateways, proxy servers, IP phones, Session Border Controllers and firewalls.

The device supports the following interfaces:

- Up to two E1/T1/J1 spans (including fractional E1/T1)
- Up to eight ISDN Basic Rate Interface (BRI) interfaces
- Up to four FXO interfaces (RJ-11 ports) - for connecting analog lines of an enterprise's PBX or the PSTN to the IP network
- Up to four FXS interfaces (RJ-11 ports) - for connecting legacy telephones, fax machines, and modems to the IP network. Optionally, the FXS interfaces can be connected to the external trunk lines of a PBX.

When deployed with a combination of FXO and FXS modules, the device can be used as a PBX for Small Office Home Office (SOHO) users, and businesses not equipped with a PBX. These interfaces can be provided in one of the following configurations:

- 1 x E1/T1 port (can support also Fractional E1/T1)
- 2 x E1/T1 ports
- 4 x BRI ports (supporting up to 8 voice calls)
- 8 x BRI ports (supporting up to 16 voice calls)
- 4 x BRI ports and 1 x E1/T1 port
- 4 x BRI ports and 4 x FXS ports
- 4 x BRI ports and 4 x FXO ports
- 4 x FXS ports and 1 x E1/T1 port
- 4 x FXO ports and 1 x E1/T1 port

The device supports various ISDN PRI protocols such as EuroISDN, North American NI2, Lucent™ 4/5ESS, Nortel™ DMS100 and others, supporting different variants of CAS protocols, including MFC R2, E&M immediate start, E&M delay dial / start, loop- and ground-start signaling. The device also supports various ISDN BRI protocols such as ETSI 5ESS and QSIG over BRI. The device also provides dual Ethernet 10/100Base-TX ports for IP redundancy.

Intelligently packaged in a stackable 1U chassis, the compact device can be mounted on a desk or in a standard 19-inch rack.

The device provides a variety of management and provisioning tools, including an HTTP-based embedded Web server, Telnet, Element Management System (EMS), and Simple Network Management Protocol (SNMP). The user-friendly, Web interface provides remote configuration using a Web browser (such as Microsoft™ Internet Explorer™).

1.3 SIP Overview

Session Initiation Protocol (SIP) is an application-layer control (signaling) protocol used on the gateway for creating, modifying, and terminating sessions with one or more participants. These sessions can include Internet telephone calls, media announcements, and conferences.

SIP invitations are used to create sessions and carry session descriptions that enable participants to agree on a set of compatible media types. SIP uses elements called Proxy servers to help route requests to the user's current location, authenticate and authorize users for services, implement provider call-routing policies and provide features to users.

SIP also provides a registration function that enables users to upload their current locations for use by Proxy servers. SIP implemented in the gateway, complies with the Internet Engineering Task Force (IETF) RFC 3261 (refer to http://www.ietf.org).

2 Configuration Concepts

You can configure the device, using the following management tools:

- The device's HTTP-based Embedded Web Server (Web interface), using any standard Web browser (described in "Web-based Management" on page 27).
- A configuration *ini* file loaded to the device (refer to "*ini* File Configuration" on page 221).
- AudioCodes' Element Management System (refer to "Element Management System (EMS)" on page 227).
- Simple Network Management Protocol (SNMP) browser software (refer to the *Product Reference Manual*).

Note: To initialize the device by assigning it an IP address, a firmware file (cmp), and a configuration file (*ini* file), you can use AudioCodes' BootP/TFTP utility, which accesses the device using its MAC address (refer to the Product Reference Manual).

Reader's Notes

3 Web-Based Management

The device's Embedded Web Server (*Web interface*) provides FCAPS (fault management, configuration, accounting, performance, and security) functionality. The Web interface allows you to remotely configure your device for quick-and-easy deployment, including uploading of software (*.cmp), configuration (*.ini), and auxiliary files, and resetting the device. The Web interface provides real-time, online monitoring of the device, including display of alarms and their severity. In addition, it displays performance statistics of voice calls and various traffic parameters.

The Web interface provides a user-friendly, graphical user interface (GUI), which can be accessed using any standard Web browser (e.g., Microsoft™ Internet Explorer). Access to the Web interface is controlled by various security mechanisms such as login user name and password, read-write privileges, and limiting access to specific IP addresses.

Notes:

- This section includes full parameter descriptions for the Web interface **configuration tables** only. For descriptions of individual parameters, refer to "Configuration Parameters Reference" on page 247.
- The Web interface allows you to configure most of the device's parameters. Those parameters that are not available in the Web interface can be configured using the *ini* file.
- Throughout this section, parameters enclosed in square brackets [...] depict the corresponding *ini* file parameters.
- Some Web interface pages are Software Upgrade Key dependant. These pages appear only if the device's installed Software Upgrade Key supports the features related to these pages. For viewing your Software Upgrade Key, refer to "Upgrading the Software Upgrade Key" on page 202.

3.1 Getting Acquainted with the Web Interface

This section describes the Web interface with regards to its graphical user interface (GUI) and basic functionality.

3.1.1 Computer Requirements

To use the device's Web interface, the following is required:

- A connection to the Internet network (World Wide Web).
- A network connection to the device's Web interface.
- One of the following Web browsers:
  - Microsoft™ Internet Explorer™ (version 6.0 or later).
  - Mozilla Firefox® (version 2.5 or later).
- Required minimum screen resolution: 1024 x 768 pixels, or 1280 x 1024 pixels.

Note: Your Web browser must be JavaScript-enabled in order to access the Web interface.

3.1.2 Accessing the Web Interface

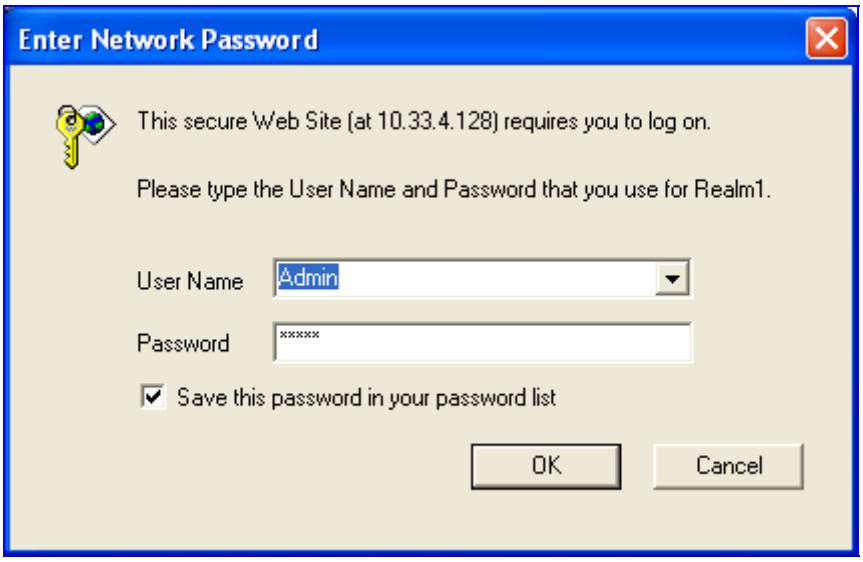
The Web interface can be opened using any standard Web browser (refer to "Computer Requirements" on page 28). When initially accessing the Web interface, use the default user name ('Admin') and password ('Admin'). For changing the login user name and password, refer to "Configuring the Web User Accounts" on page 83).

Note: For assigning an IP address to the device, refer to the device's *Installation Manual*.

➢ **To access the Web interface:**

1. Open a standard Web browser application.
2. In the Web browser's Uniform Resource Locator (URL) field, specify the device's IP address (e.g., http://10.1.10.10); the Web interface's 'Enter Network Password' dialog box appears, as shown in the figure below**:**

Figure 3-1: Enter Network Password Screen


Enter Network Password
This secure Web Site (at 10.33.4.128) requires you to log on.
Please type the User Name and Password that you use for Realm1.
User Name
Admin
Password
*****
Save this password in your password list
OK
Cancel

3. In the 'User Name' and 'Password' fields, enter the case-sensitive, user name and password.
4. Click the **OK** button; the Web interface is accessed, displaying the 'Home' page (for a detailed description of the 'Home' page, refer to "Using the Home Page" on page 54).

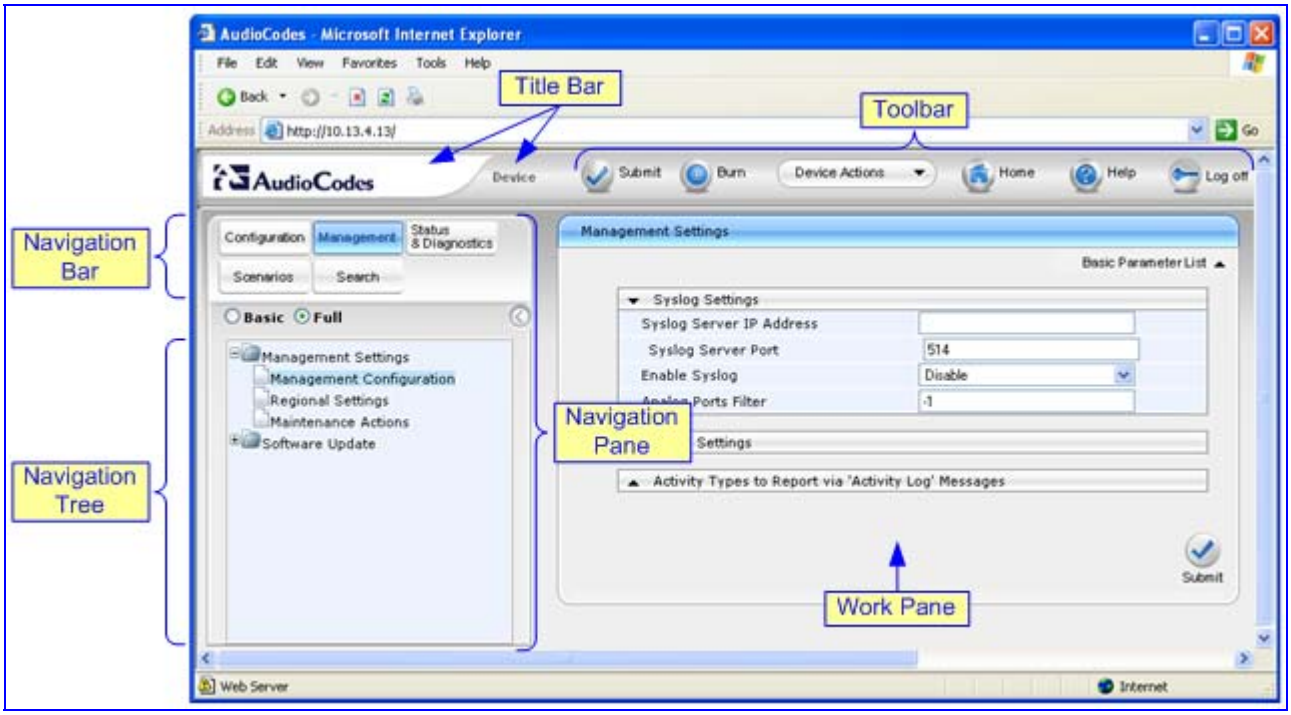
Note: If access to the device's Web interface is denied ("Unauthorized") due to Microsoft Internet Explorer security settings, perform the following troubleshooting procedures:

1. Delete all cookies in the Temporary Internet Files folder. If this does not resolve the problem, the security settings may need to be altered (continue with Step 2).
2. In Internet Explorer, navigate to **Tools** menu > **Internet Options** > **Security** tab > **Custom Level**, and then scroll down to the Logon options and select **Prompt for username and password**. Select the **Advanced** tab, and then scroll down until the HTTP 1.1 Settings are displayed and verify that **Use HTTP 1.1** is selected.
3. Quit and start the Web browser again.

3.1.3 Areas of the GUI

The figure below displays the general layout of the Graphical User Interface (GUI) of the Web interface:

Figure 3-2: Main Areas of the Web Interface GUI



The Web GUI is composed of the following main areas:

- **Title bar:** Displays the corporate logo and product name.
- **Toolbar:** Provides frequently required command buttons for configuration (refer to "Toolbar" on page 31).
- **Navigation Pane:** Consists of the following areas:
  - **Navigation bar:** Provides tabs for accessing the configuration menus (refer to "Navigation Tree" on page 32), creating a Scenario (refer to Scenarios on page 42), and searching *ini* file parameters that have corresponding Web interface parameters (refer to "Searching for Configuration Parameters" on page 41).
  - **Navigation tree:** Displays the elements pertaining to the tab selected on the Navigation bar (tree-like structure of the configuration menus, Scenario Steps, or Search engine).
- **Work pane:** Displays configuration pages where all configuration is performed (refer to "Working with Configuration Pages" on page 34).

3.1.4 Toolbar

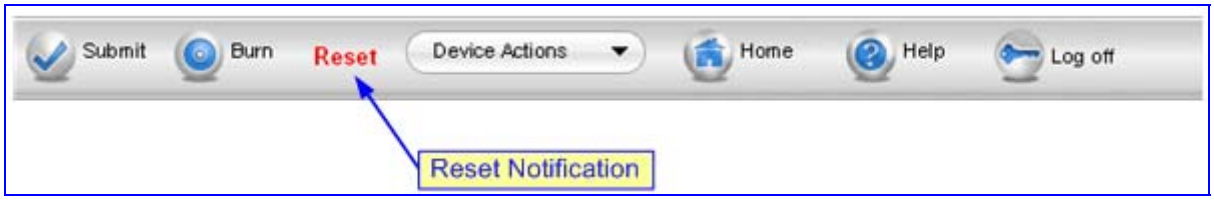
The toolbar provides command buttons for quick-and-easy access to frequently required commands, as described in the table below:

Table 3-1: Description of Toolbar Buttons

Icon	Button Name	Description
	Submit	Applies parameter settings to the device (refer to "Saving Configuration" on page 199). **Note:** This icon is grayed out when not applicable to the currently opened page.
	Burn	Saves parameter settings to flash memory (refer to "Saving Configuration" on page 199).
	Device Actions	Opens a drop-down menu list with frequently needed commands: ▪ **Load Configuration File:** opens the 'Configuration File' page for loading an *ini* file (refer to "Backing Up and Restoring Configuration" on page 208). ▪ **Save Configuration File:** opens the 'Configuration File' page for saving the *ini* file to a PC (refer to "Backing Up and Restoring Configuration" on page 208). ▪ **Reset:** opens the 'Maintenance Actions' page for resetting the device (refer to "Resetting the Device" on page 197). ▪ **Software Upgrade Wizard:** opens the 'Software Upgrade Wizard' page for upgrading the device's software (refer to "Software Upgrade Wizard" on page 205).
	Home	Opens the 'Home' page (refer to "Using the Home Page" on page 54).
	Help	Opens the Online Help topic of the currently opened configuration page in the Work pane (refer to "Getting Help" on page 51).
	Log off	Logs off a session with the Web interface (refer to "Logging Off the Web Interface" on page 52).

Note: If you modify parameters that take effect only after a device reset, after you click the **Submit** button, the toolbar displays the word "Reset" (in red color), as shown in the figure below. This is a reminder to later save ('burn') your settings to flash memory and reset the device.

Figure 3-3: "Reset" Displayed on Toolbar



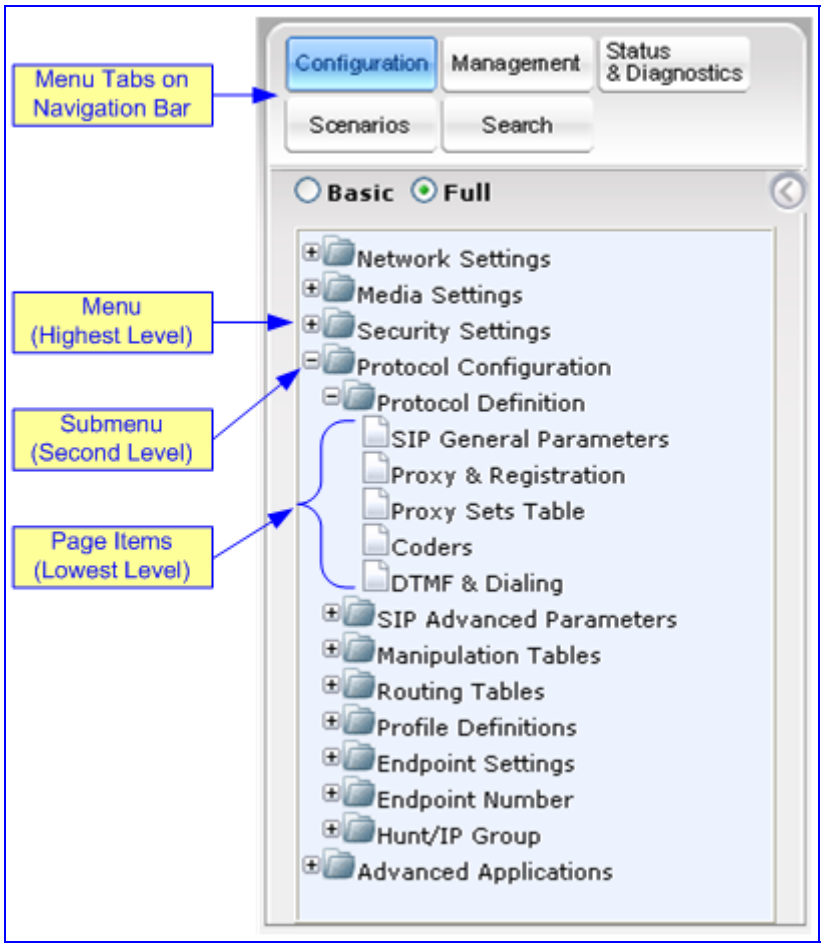
3.1.5 Navigation Tree

The Navigation tree, located in the Navigation pane, displays the menus (pertaining to the menu tab selected on the Navigation bar) used for accessing the configuration pages. The Navigation tree displays a tree-like structure of menus. You can easily drill-down to the required page item level to open its corresponding page in the Work pane.

The terminology used throughout this manual for referring to the hierarchical structure of the tree is as follows:

- *menu*: first level (highest level)
- *submenu*: second level - contained within a menu.
- *page item*: last level (lowest level in a menu) - contained within a menu or submenu.

Figure 3-4: Terminology for Navigation Tree Levels



➢ To view menus in the Navigation tree:

- On the Navigation bar, select the required tab:
  - **Configuration** (refer to "Configuration Tab" on page 60)
  - **Management** (refer to "Management Tab" on page 188)
  - **Status & Diagnostics** (refer to "Status & Diagnostics Tab" on page 209)

➢ **To navigate to a page:**

1. Navigate to the required page item, by performing the following:
   - Drilling-down using the **plus** ⊞ signs to expand the menus and submenus
   - Drilling-up using the **minus** ⊟ signs to collapse the menus and submenus
2. Select the required page item; the page opens in the Work pane.

3.1.5.1 Displaying Navigation Tree in Basic and Full View

You can view an expanded or reduced Navigation tree display regarding the number of listed menus and submenus. This is relevant when using the configuration tabs (**Configuration**, **Management**, and **Status & Diagnostics**) on the Navigation bar.

The Navigation tree menu can be displayed in one of two views:

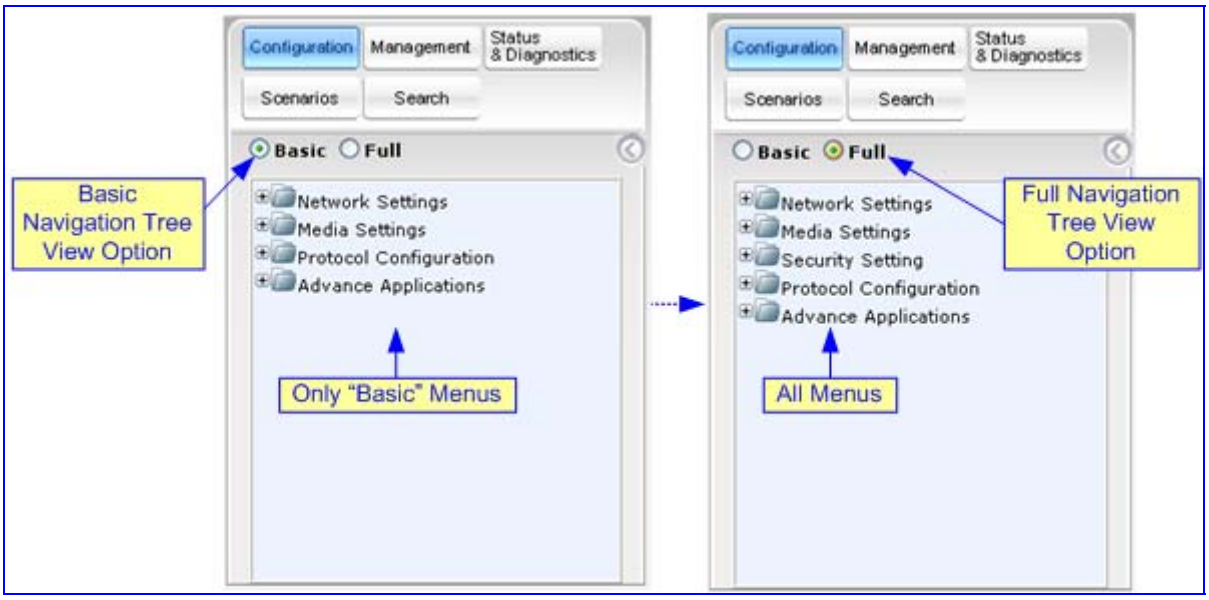
- **Basic:** displays only commonly used menus
- **Full:** displays all the menus pertaining to a configuration tab.

The advantage of the Basic view is that it prevents "cluttering" the Navigation tree with menus that may not be required. Therefore, a Basic view allows you to easily locate required menus.

➢ **To toggle between Full and Basic view:**

- Select the **Basic** option (located below the Navigation bar) to display a reduced menu tree; select the **Full** option to display all the menus. By default, the **Basic** option is selected.

Figure 3-5: Navigation Tree in Basic and Full View



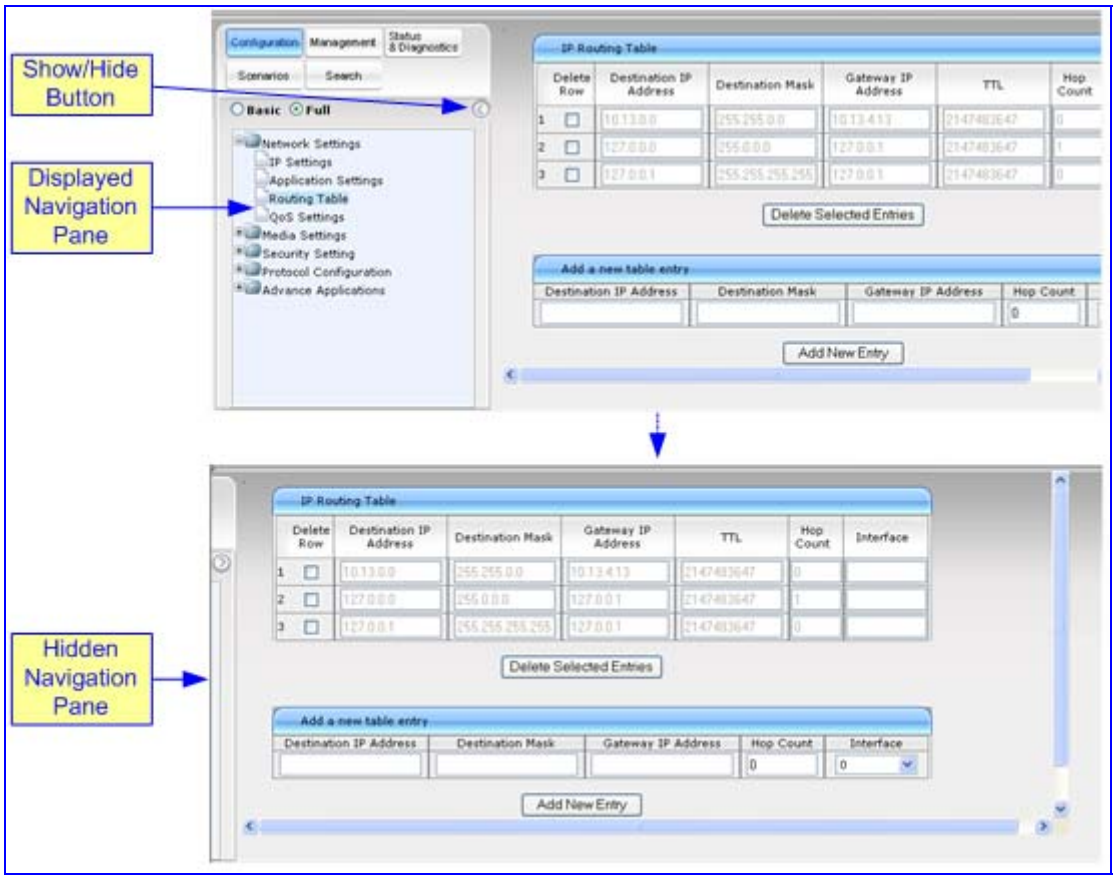
Note: When in Scenario mode (refer to Scenarios on page 42), the Navigation tree is displayed in 'Full' view (i.e., all menus are displayed in the Navigation tree).

3.1.5.2 Showing / Hiding the Navigation Pane

The Navigation pane can be hidden to provide more space for elements displayed in the Work pane. This is especially useful when the Work pane displays a page with a table that's wider than the Work pane and to view the all the columns, you need to use scroll bars. The arrow button located just below the Navigation bar is used to hide and show the Navigation pane.

- **To hide the Navigation pane:** click the left-pointing arrow ; the pane is hidden and the button is replaced by the right-pointing arrow button.
- **To show the Navigation pane:** click the right-pointing arrow ; the pane is displayed and the button is replaced by the left-pointing arrow button.

Figure 3-6: Showing and Hiding Navigation Pane



3.1.6 Working with Configuration Pages

The configuration pages contain the parameters for configuring the device. The configuration pages are displayed in the Work pane, which is located to the right of the Navigation pane.

3.1.6.1 Accessing Pages

The configuration pages are accessed by clicking the required page item in the Navigation tree.

➢ **To open a configuration page in the Work pane:**

1. On the Navigation bar, click the required tab:
   - **Configuration** (refer to "Configuration Tab" on page 60)
   - **Management** (refer to "Management Tab" on page 188)
   - **Status & Diagnostics** (refer to "Status & Diagnostics Tab" on page 209)

   The menus of the selected tab appears in the Navigation tree.
2. In the Navigation tree, drill-down to the required page item; the page opens in the Work pane.

You can also access previously opened pages, by clicking your Web browser's **Back** button until you have reached the required page. This is useful if you want to view pages in which you have performed configurations in the current Web session.

Notes:

- You can also access certain pages from the **Device Actions** button located on the toolbar (refer to "Toolbar" on page 31).
- To view all the menus in the Navigation tree, ensure that the Navigation tree is in 'Full' view (refer to "Displaying Navigation Tree in Basic and Full View" on page 33).
- To get Online Help for the currently opened page, refer to "Getting Help" on page 51.
- Certain pages may not be accessible if your Web user account's access level is low (refer to "Configuring the Web User Accounts" on page 83).

3.1.6.2 Viewing Parameters

For convenience, some pages allow you to view a reduced or expanded display of parameters. A reduced display allows you to easily identify required parameters, enabling you to quickly configure your device.

The Web interface provides you with two methods for handling the display of page parameters:

- Display of "basic" and "advanced" parameters (refer to "Displaying Basic and Advanced Parameters" on page 36)
- Display of parameter groups (refer to "Showing / Hiding Parameter Groups" on page 37)

Note: Certain pages may only be read-only if your Web user account's access level is low (refer to "Configuring the Web User Accounts" on page 83). If a page is read-only, 'Read-Only Mode' is displayed at the bottom of the page.

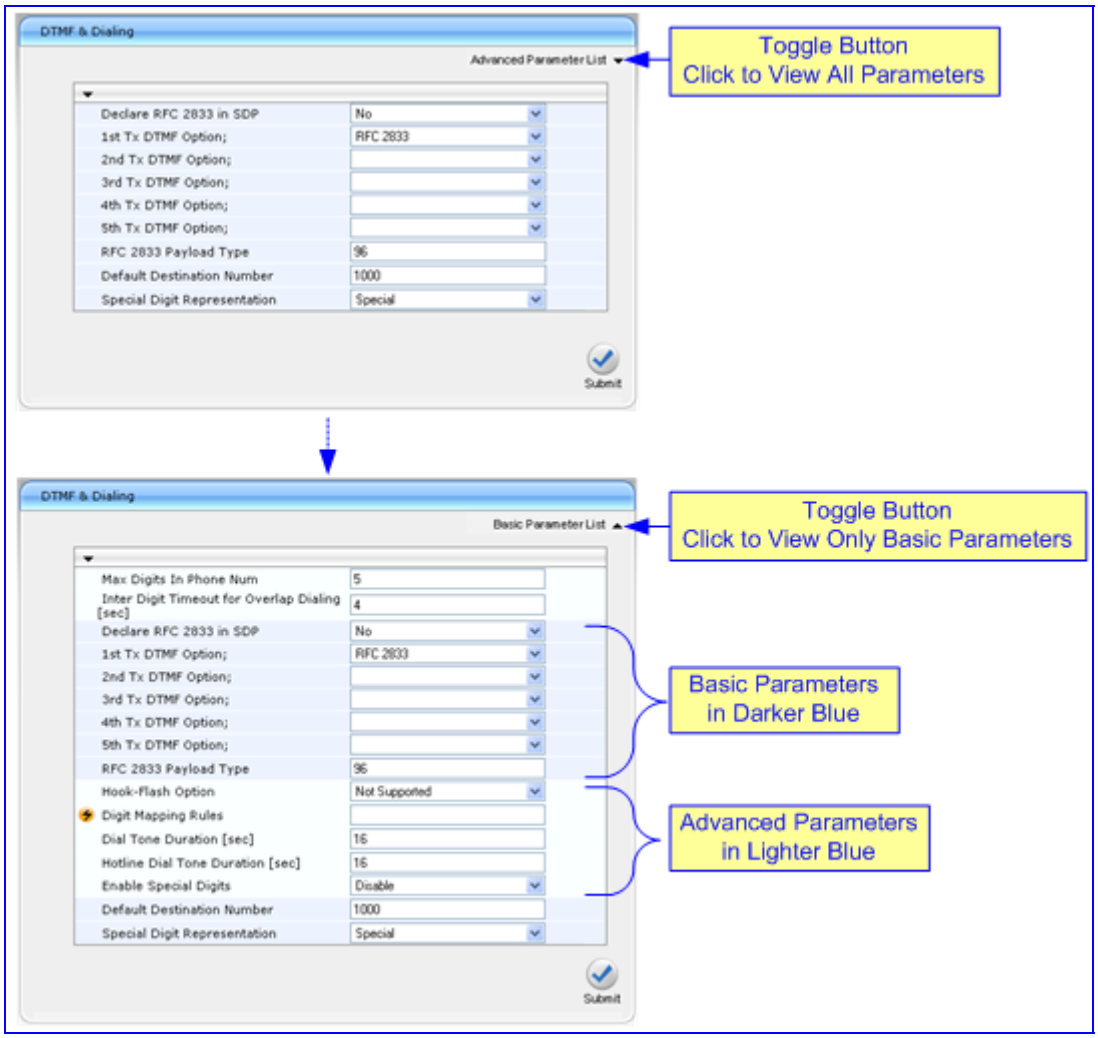
3.1.6.2.1 Displaying Basic and Advanced Parameters

Some pages provide you with an **Advanced Parameter List** / **Basic Parameter List** toggle button that allows you to show or hide advanced parameters (in addition to displaying the basic parameters). This button is located on the top-right corner of the page and has two states:

- **Advanced Parameter List** button with down-pointing arrow: click this button to display all parameters.
- **Basic Parameter List** button with up-pointing arrow: click this button to show only common (*basic*) parameters.

The figure below shows an example of a page displaying basic parameters only, and then showing advanced parameters as well, using the **Advanced Parameter List** button.

Figure 3-7: Toggling between Basic and Advanced Page View



For ease of identification, the basic parameters are displayed with a darker blue color background than the advanced parameters.

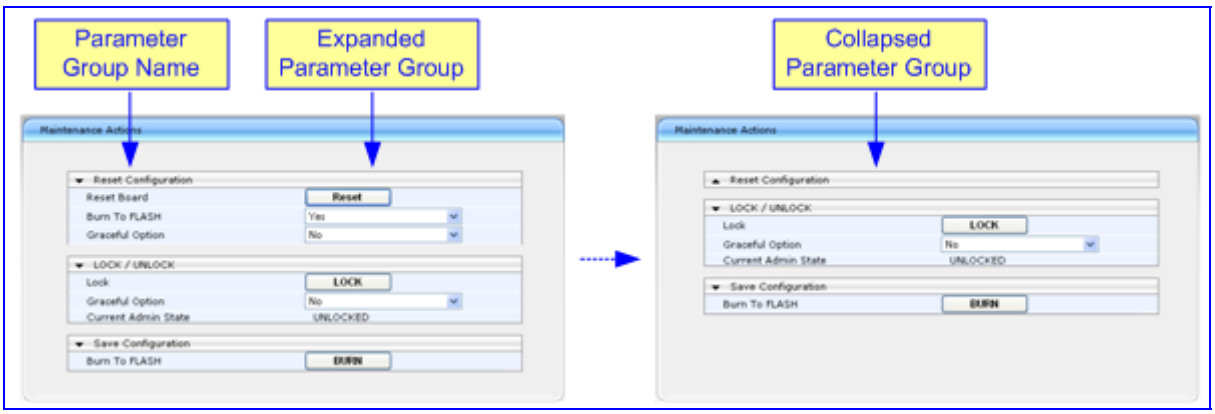
Notes:

- When the Navigation tree is in 'Full' mode (refer to "Navigation Tree" on page 32), configuration pages display all their parameters (i.e., the 'Advanced Parameter List' view is displayed).
- If a page contains only basic parameters, the **Basic Parameter List** button is not displayed.

3.1.6.2.2 Showing / Hiding Parameter Groups

Some pages provide groups of parameters, which can be hidden or shown. To toggle between hiding and showing a group, simply click the group name button that appears above each group. The button appears with a down-pointing or up-pointing arrow, indicating that it can be collapsed or expanded when clicked, respectively.

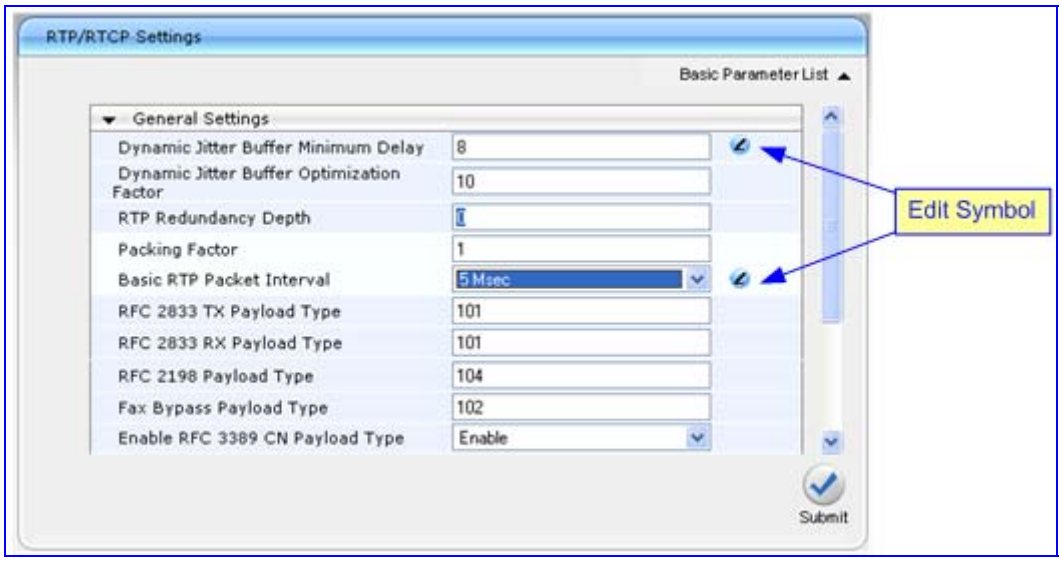
Figure 3-8: Expanding and Collapsing Parameter Groups



3.1.6.3 Modifying and Saving Parameters

When you change parameter values on a page, the **Edit** symbol appears to the right of these parameters. This is especially useful for indicating the parameters that you have currently modified (before applying the changes). After you save your parameter modifications (refer to the procedure described below), the **Edit** symbols disappear.

Figure 3-9: Editing Symbol after Modifying Parameter Value



➢ **To save configuration changes on a page to the device's volatile memory (RAM):**

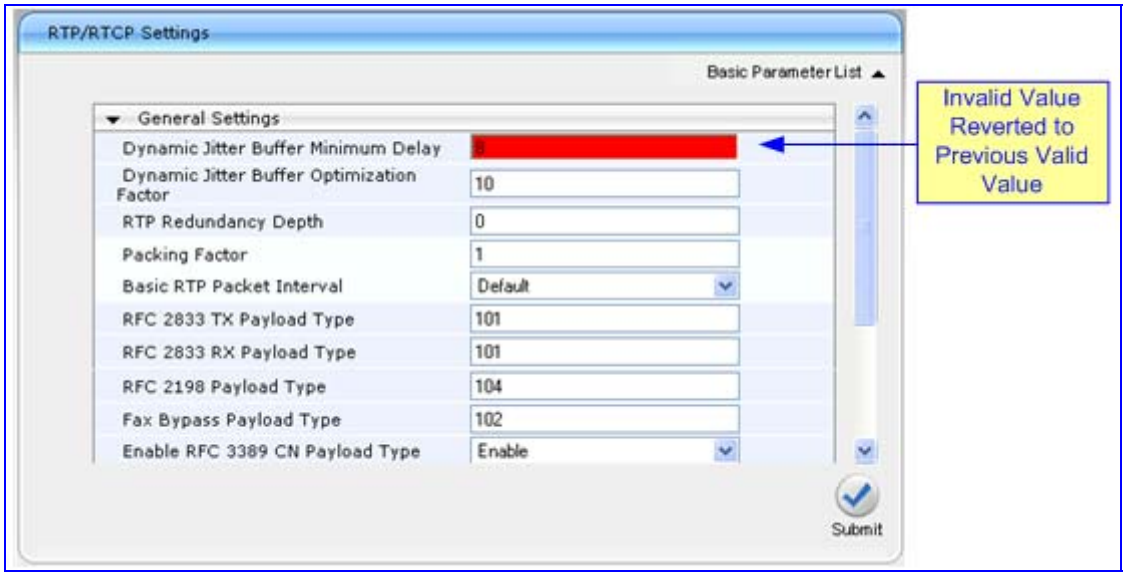
- Click the **Submit** button, which is located near the bottom of the page in which you are working; modifications to parameters with on-the-fly capabilities are immediately applied to the device and take effect; other parameters (displayed on the page with the lightning symbol) are not changeable on-the-fly and require a device reset (refer to "Resetting the Device" on page 197) before taking effect.

Notes:

- Parameters saved to the volatile memory (by clicking **Submit**), revert to their previous settings after a hardware or software reset (or if the device is powered down). Therefore, to ensure parameter changes (whether on-the-fly or not) are retained, you need to save ('burn') them to the device's non-volatile memory, i.e., flash (refer to "Saving Configuration" on page 199).
- If you modify a parameter value and then attempt to navigate away from the page without clicking **Submit**, a message box appears notifying you of this. Click **Yes** to save your modifications or **No** to ignore them.

If you enter an invalid parameter value (e.g., not in the range of permitted values) and then click **Submit**, a message box appears notifying you of the invalid value. In addition, the parameter value reverts to its previous value and is highlighted in red, as shown in the figure below:

Figure 3-10: Value Reverts to Previous Valid Value



3.1.6.4 Entering Phone Numbers

Phone numbers or prefixes that you need to configure throughout the Web interface must be entered only as digits without any other characters. For example, if you wish to enter the phone number 555-1212, it must be entered as 5551212 without the hyphen (-). If the hyphen is entered, the entry is invalid.

3.1.6.5 Working with Tables

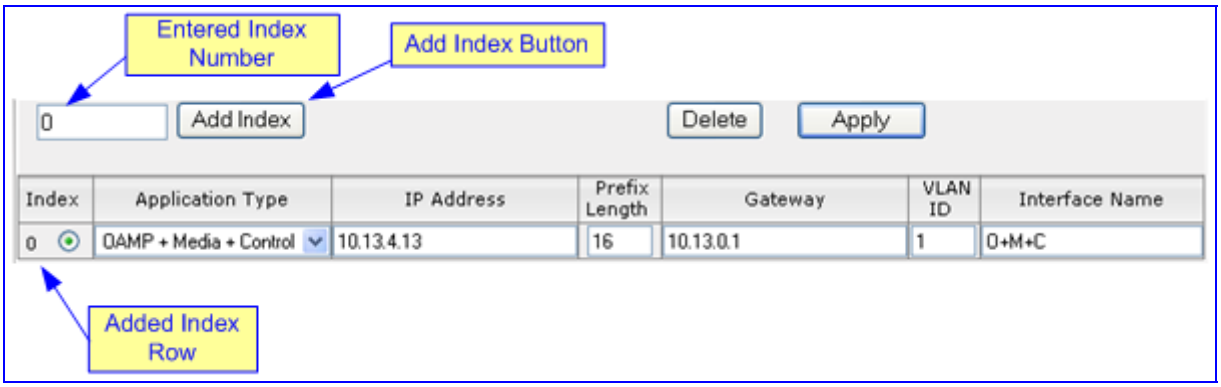
The Web interface includes many configuration pages that provide tables for configuring the device. Some of these tables provide the following command buttons:

- **Add Index:** adds an index entry to the table.
- **Duplicate:** duplicates a selected, existing index entry.
- **Compact:** organizes the index entries in ascending, consecutive order.
- **Delete:** deletes a selected index entry.
- **Apply:** saves the configuration.

➢ **To add an entry to a table:**

1. In the 'Add Index' field, enter the desired index entry number, and then click **Add Index**; an index entry row appears in the table:

Figure 3-11: Adding an Index Entry to a Table



2. Click **Apply** to save the index entry.

Notes:

- Before you can add another index entry, you must ensure that you have applied the previously added index entry (by clicking **Apply**).
- If you leave the 'Add' field blank and then click **Add Index**, the existing index entries are all incremented by one and the newly added index entry is assigned the index 0.

➢ **To add a copy of an existing index table entry:**

1. In the 'Index' column, select the index that you want to duplicate; the **Edit** button appears.
2. Click **Edit**; the fields in the corresponding index row become available.
3. Click **Duplicate**; a new index entry is added with identical settings as the selected index in Step 1. In addition, all existing index entries are incremented by one and the newly added index entry is assigned the index 0.

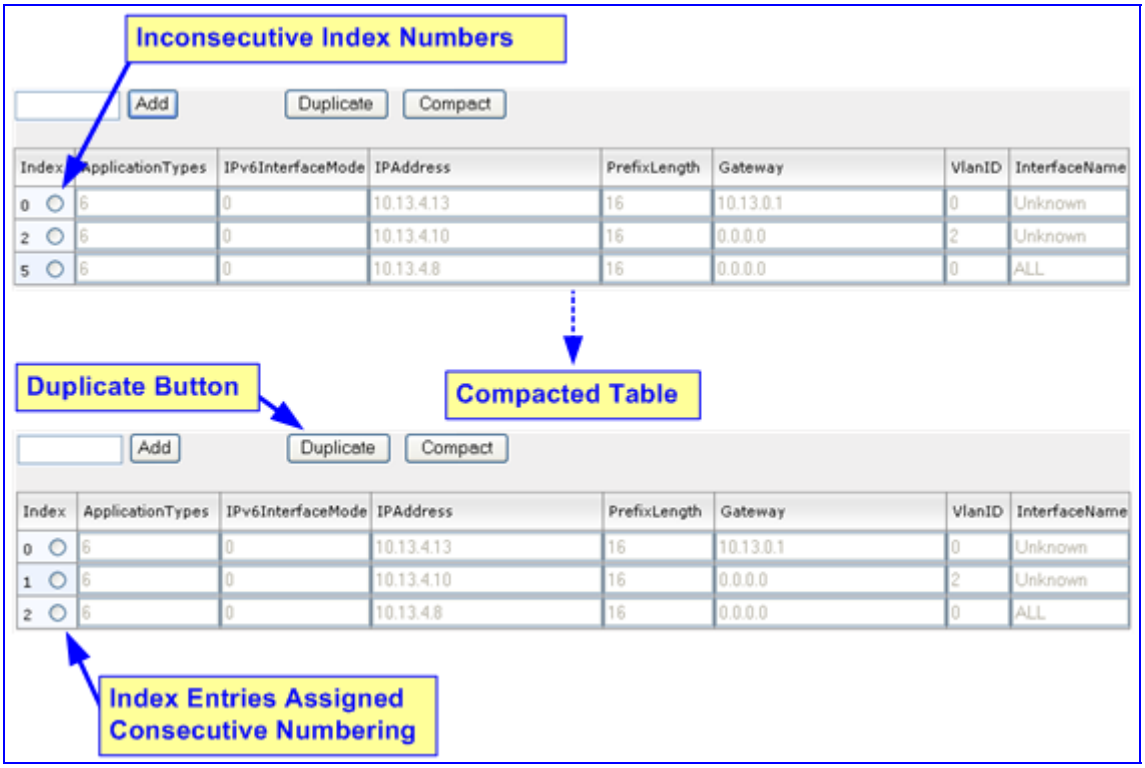
➢ **To edit an existing index table entry:**

1. In the 'Index' column, select the index corresponding to the table row that you want to edit.
2. Click **Edit**; the fields in the corresponding index row become available.
3. Modify the values as required, and then click **Apply**; the new settings are applied.

➢ **To organize the index entries in ascending, consecutive order:**

- Click **Compact**; the index entries are organized in ascending, consecutive order, starting from index 0. For example, if you added three index entries 0, 4, and 6, then the index entry 4 is re-assigned index number 1 and the index entry 6 is re-assigned index number 2.

Figure 3-12: Compacting a Web Interface Table



➢ **To delete an existing index table entry:**

1. In the 'Index' column, select the index corresponding to the table row that you want to delete.
2. Click **Delete**; the table row is removed from the table.

3.1.7 Searching for Configuration Parameters

The Web interface provides a search engine that allows you to search any *ini* file parameter that is configurable by the Web interface (i.e., has a corresponding Web parameter). You can search for a specific parameter (e.g., "EnableIPSec") or a sub-string of that parameter (e.g., "sec"). If you search for a sub-string, all parameters that contain the searched sub-string in their names are listed.

➢ **To search for *ini* file parameters configurable in the Web interface:**

1. On the Navigation bar, click the **Search** tab; the Search engine appears in the Navigation pane.

2. In the 'Search' field, enter the parameter name or sub-string of the parameter name that you want to search. If you have performed a previous search for such a parameter, instead of entering the required string, you can use the 'Search History' drop-down list to select the string (saved from a previous search).

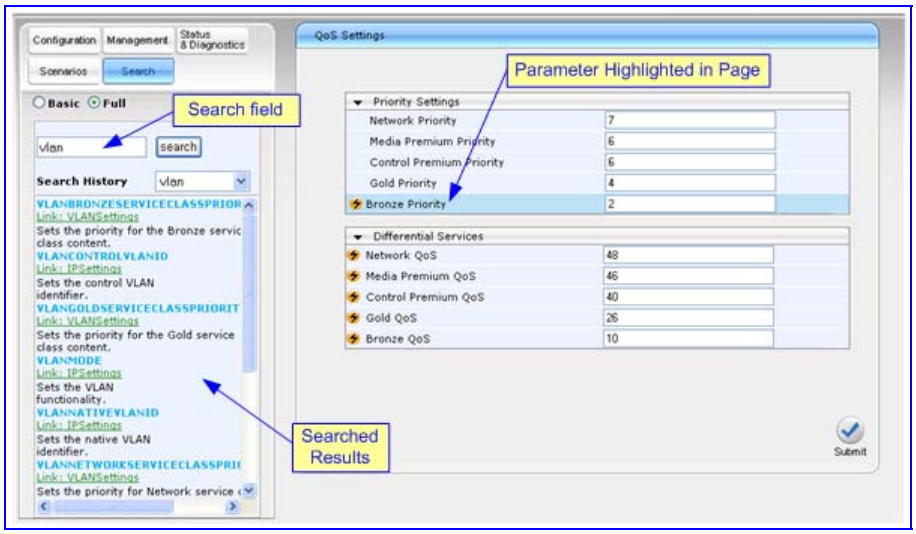
3. Click **Search**; a list of located parameters based on your search appears in the Navigation pane.

   Each searched result displays the following:

   - *ini* file parameter name
   - Link (in green) to its location (page) in the Web interface
   - Brief description of the parameter

4. In the searched list, click the required parameter (link in green) to open the page in which the parameter appears; the relevant page opens in the Work pane and the searched parameter is highlighted for easy identification, as shown in the figure below:

Figure 3-13: Searched Result Screen



Note: If the searched parameter is not located, a notification message is displayed.

3.1.8 Working with Scenarios

The Web interface allows you to create your own "menu" with up to 20 pages selected from the menus in the Navigation tree (i.e., pertaining to the **Configuration**, **Management**, and **Status & Diagnostics** tabs). The "menu" is a set of configuration pages grouped into a logical entity referred to as a *Scenario*. Each page in the Scenario is referred to as a *Step*. For each Step, you can select up to 25 parameters in the page that you want available in the Scenario. Therefore, the Scenario feature is useful in that it allows you quick-and-easy access to commonly used configuration parameters specific to your network environment. When you login to the Web interface, your Scenario is displayed in the Navigation tree, thereby, facilitating your configuration.

Instead of creating a Scenario, you can also load an existing Scenario from a PC to the device (refer to "Loading a Scenario to the Device" on page 48).

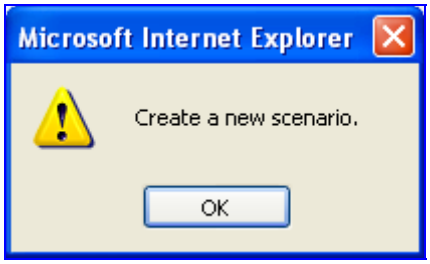
3.1.8.1 Creating a Scenario

The Web interface allows you to create one Scenario with up to 20 configuration pages, as described in the procedure below:

➢ **To create a Scenario:**

1. On the Navigation bar, click the **Scenarios** tab; a message box appears, requesting you to confirm creation of a Scenario:

Figure 3-14: Scenario Creation Confirm Message Box



Note: If a Scenario already exists, the Scenario Loading message box appears.

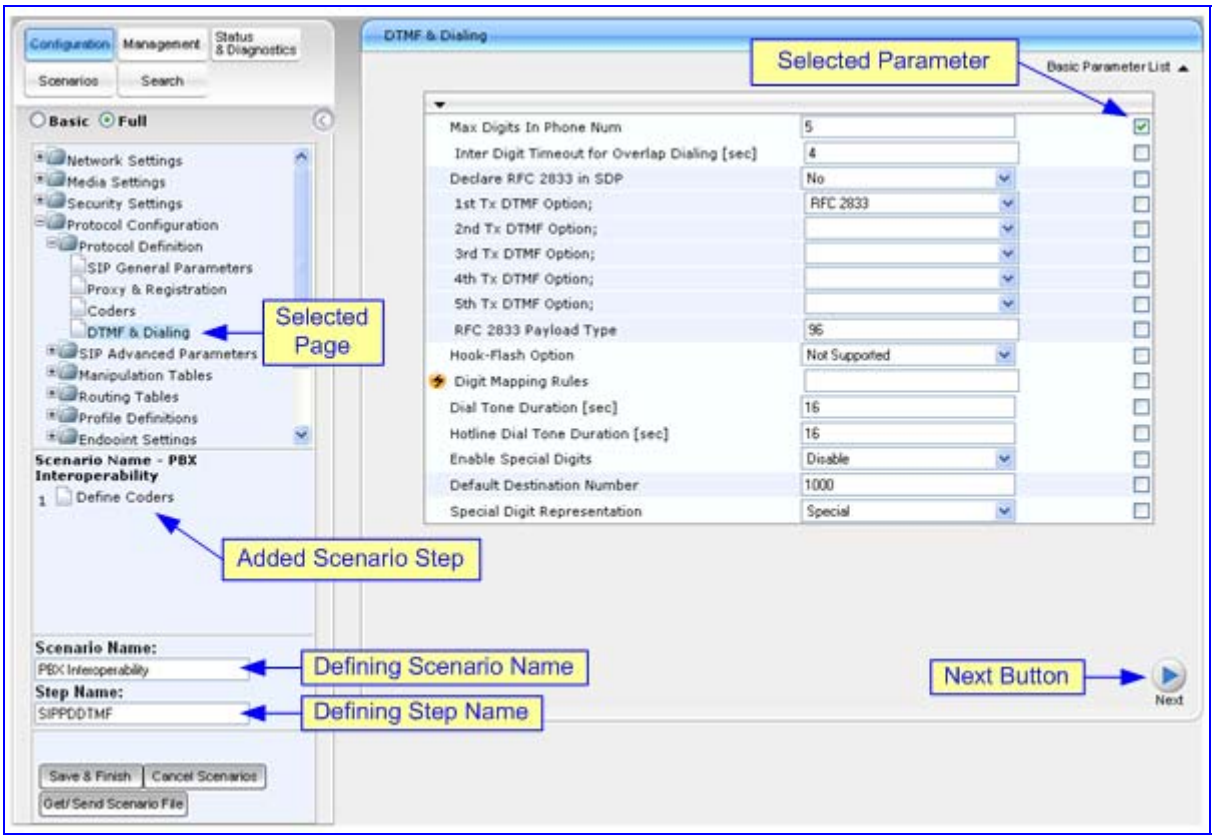
2. Click **OK**; the Scenario mode appears in the Navigation tree as well as the menus of the **Configuration** tab.

Note: If a Scenario already exists and you wish to create a new one, click the **Create Scenario** button, and then click **OK** in the subsequent message box.

3. In the 'Scenario Name' field, enter an arbitrary name for the Scenario.
4. On the Navigation bar, click the **Configuration** or **Management** tab to display their respective menus in the Navigation tree.
5. In the Navigation tree, select the required page item for the Step, and then in the page itself, select the required parameters by selecting the check boxes corresponding to the parameters.
6. In the 'Step Name' field, enter a name for the Step.

7. Click the **Next** button located at the bottom of the page; the Step is added to the Scenario and appears in the Scenario Step list:

Figure 3-15: Creating a Scenario



8. Repeat steps 5 through 8 to add additional Steps (i.e., pages).

9. When you have added all the required Steps for your Scenario, click the **Save & Finish** button located at the bottom of the Navigation tree; a message box appears informing you that the Scenario has been successfully created.

10. Click **OK**; the Scenario mode is quit and the menu tree of the **Configuration** tab appears in the Navigation tree.

Notes:

- You can add up to 20 Steps to a Scenario, where each Step can contain up to 25 parameters.
- When in Scenario mode, the Navigation tree is in 'Full' display (i.e., all menus are displayed in the Navigation tree) and the configuration pages are in 'Advanced Parameter List' display (i.e., all parameters are shown in the pages). This ensures accessibility to all parameters when creating a Scenario. For a description on the Navigation tree views, refer to "Navigation Tree" on page 32.
- If you previously created a Scenario and you click the **Create Scenario** button, the previously created Scenario is deleted and replaced with the one you are creating.
- Only users with access level of 'Security Administrator' can create a Scenario.

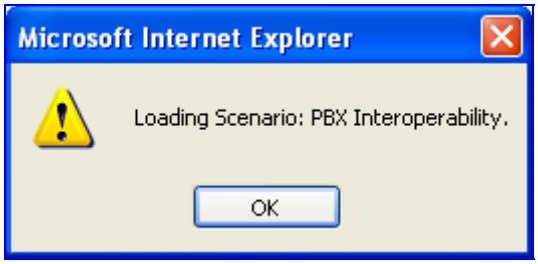
3.1.8.2 Accessing a Scenario

Once you have created the Scenario, you can access it at anytime by following the procedure below:

➢ **To access the Scenario:**

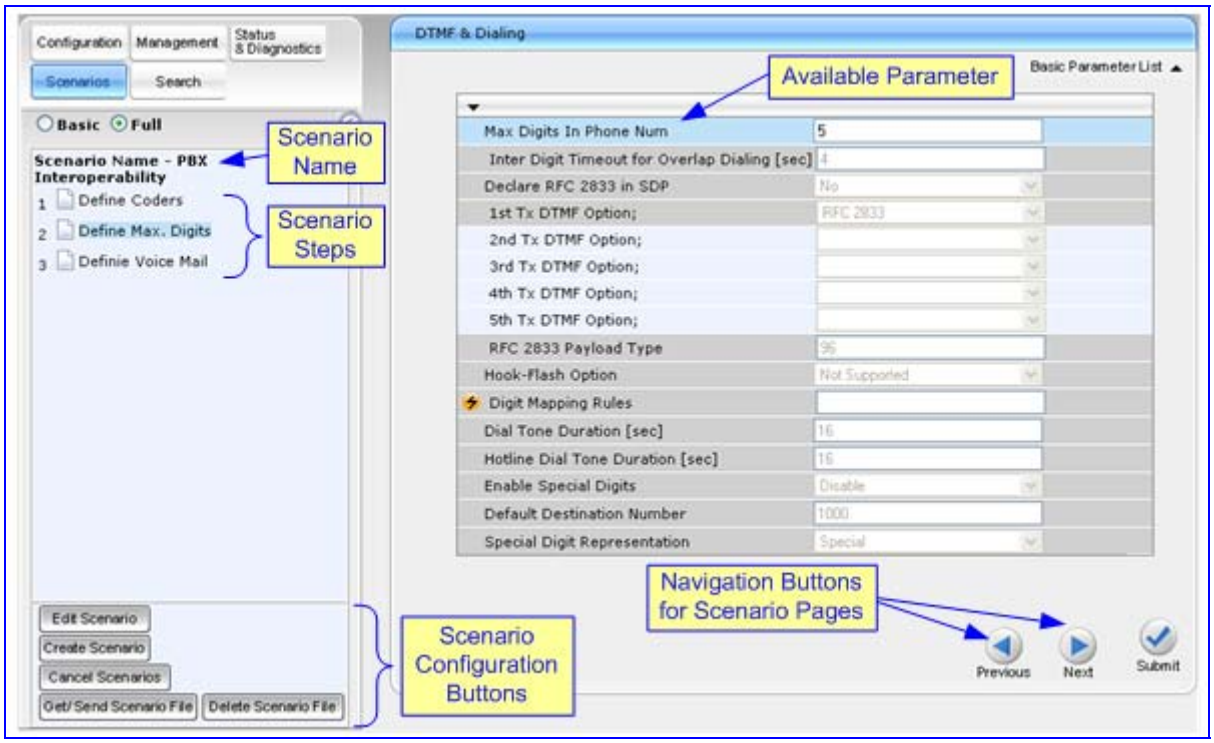
1. On the Navigation bar, select the **Scenario** tab; a message box appears, requesting you to confirm the loading of the Scenario.

Figure 3-16: Scenario Loading Message Box



2. Click **OK**; the Scenario and its Steps appear in the Navigation tree, as shown in the example figure below:

Figure 3-17: Scenario Example



When you select a Scenario Step, the corresponding page is displayed in the Work pane. In each page, the available parameters are indicated by a dark-blue background; the unavailable parameters are indicated by a gray or light-blue background.

To navigate between Scenario Steps, you can perform one of the following:

- In the Navigation tree, click the required Scenario Step.
- In an opened Scenario Step (i.e., page appears in the Work pane), use the following navigation buttons:
  - **Next:** opens the next Step listed in the Scenario.
  - **Previous**: opens the previous Step listed in the Scenario.

Note: If you reset the device while in Scenario mode, after the device resets, you are returned once again to the Scenario mode.

3.1.8.3 Editing a Scenario

You can modify a Scenario anytime by adding or removing Steps (i.e., pages) or parameters, and changing the Scenario name and the Steps' names.

Note: Only users with access level of 'Security Administrator' can edit a Scenario.

➢ **To edit a Scenario:**

1. On the Navigation bar, click the **Scenarios** tab; a message box appears, requesting you to confirm Scenario loading.
2. Click **OK**; the Scenario appears with its Steps in the Navigation tree.
3. Click the **Edit Scenario** button located at the bottom of the Navigation pane; the 'Scenario Name' and 'Step Name' fields appear.
4. You can perform the following edit operations:
   - **Add Steps:**
     a. On the Navigation bar, select the desired tab (i.e., **Configuration** or **Management**); the tab's menu appears in the Navigation tree.
     b. In the Navigation tree, navigate to the desired page item; the corresponding page opens in the Work pane.
     c. In the page, select the required parameter(s) by marking the corresponding check box(es).
     d. Click **Next**.

- **Add or Remove Parameters:**
  a. In the Navigation tree, select the required Step; the corresponding page opens in the Work pane.
  b. To add parameters, select the check boxes corresponding to the desired parameters; to remove parameters, clear the check boxes corresponding to the parameters that you want removed.
  c. Click **Next**.
- **Edit the Step Name:**
  a. In the Navigation tree, select the required Step.
  b. In the 'Step Name' field, modify the Step name.
  c. In the page, click **Next**.
- **Edit the Scenario Name:**
  a. In the 'Scenario Name' field, edit the Scenario name.
  b. In the displayed page, click **Next**.
- **Remove a Step:**
  a. In the Navigation tree, select the required Step; the corresponding page opens in the Work pane.
  b. In the page, clear all the check boxes corresponding to the parameters.
  c. Click **Next**.

5. After clicking **Next**, a message box appears notifying you of the change. Click **OK**.
6. Click **Save & Finish**; a message box appears informing you that the Scenario has been successfully modified. The Scenario mode is exited and the menus of the **Configuration** tab appear in the Navigation tree.

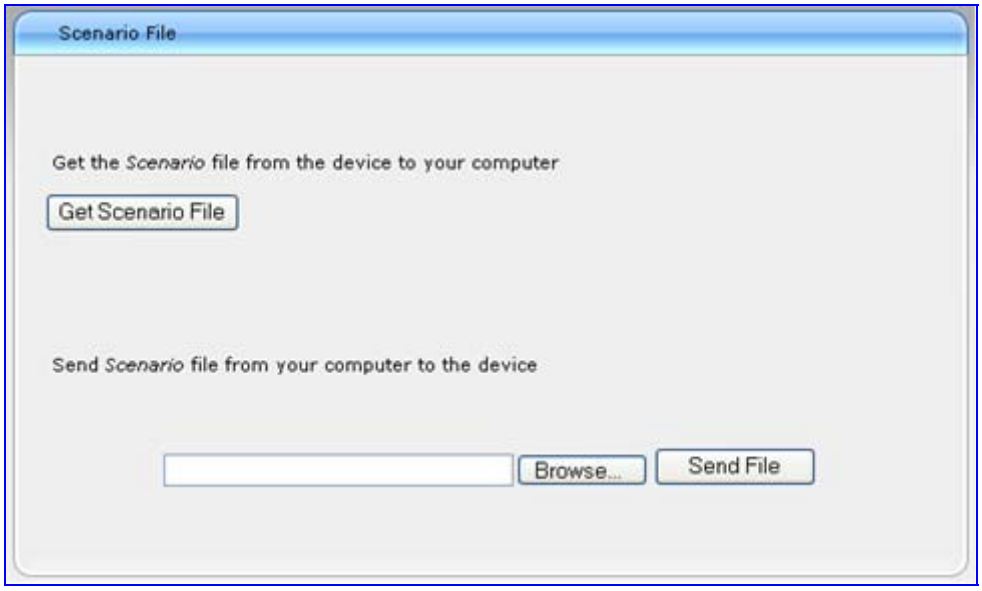
3.1.8.4 Saving a Scenario to a PC

You can save a Scenario to a PC (as a *dat* file). This is especially useful when requiring more than one Scenario to represent different environment setups (e.g., where one includes PBX interoperability and another not). Once you create a Scenario and save it to your PC, you can then keep on saving modifications to it under different Scenario file names. When you require a specific network environment setup, you can simply load the suitable Scenario file from your PC (refer to "Loading a Scenario to the Device" on page 48).

➢ **To save a Scenario to a PC:**

1. On the Navigation bar, click the **Scenarios** tab; the Scenario appears in the Navigation tree.
2. Click the **Get/Send Scenario File** button (located at the bottom of the Navigation tree); the 'Scenario File' page appears, as shown below:

Figure 3-18: Scenario File Page


Scenario File
Get the Scenario file from the device to your computer
Get Scenario File
Send Scenario file from your computer to the device
Browse...
Send File

3. Click the **Get Scenario File** button; the 'File Download' window appears.
4. Click **Save**, and then in the 'Save As' window navigate to the folder to where you want to save the Scenario file. When the file is successfully downloaded to your PC, the 'Download Complete' window appears.
5. Click **Close** to close the 'Download Complete' window.

3.1.8.5 Loading a Scenario to the Device

Instead of creating a Scenario, you can load a Scenario file (*data* file) from your PC to the device.

➢ **To load a Scenario to the device:**

1. On the Navigation bar, click the **Scenarios** tab; the Scenario appears in the Navigation tree.
2. Click the **Get/Send Scenario File** button (located at the bottom of the Navigation tree); the 'Scenario File' page appears (refer to "Saving a Scenario to a PC" on page 47).

3. Click the **Browse** button, and then navigate to the Scenario file stored on your PC.
4. Click the **Send File** button.

Notes:

- You can only load a Scenario file to a device that has an identical hardware configuration setup to the device in which it was created. For example, if the Scenario was created in a device with FXS interfaces, the Scenario cannot be loaded to a device that does not have FXS interfaces.
- The loaded Scenario replaces any existing Scenario.
- You can also load a Scenario file using BootP, by loading an *ini* file that contains the *ini* file parameter ScenarioFileName (refer to Web and Telnet Parameters on page 260). The Scenario dat file must be located in the same folder as the *ini* file. For a detailed description on BootP, refer to the Product Reference Manual.

3.1.8.6 Deleting a Scenario

You can delete the Scenario by using the **Delete Scenario File** button, as described in the procedure below:

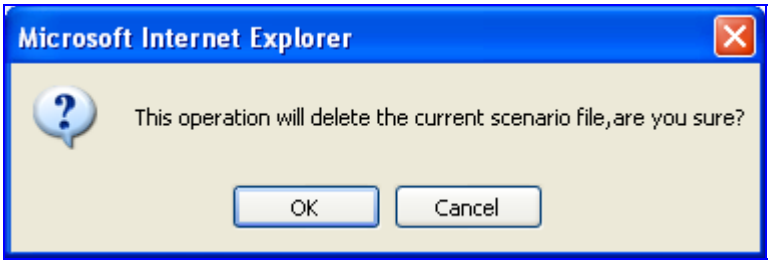
➢ **To delete the Scenario:**

1. On the Navigation bar, click the **Scenarios** tab; a message box appears, requesting you to confirm:

Figure 3-19: Scenario Loading Message Box

2. Click **OK**; the Scenario mode appears in the Navigation tree.
3. Click the **Delete Scenario File** button; a message box appears requesting confirmation for deletion.

Figure 3-20: Message Box for Confirming Scenario Deletion



4. Click **OK**; the Scenario is deleted and the Scenario mode closes.

Note: You can also delete a Scenario using the following alternative methods:

- Loading an empty *dat* file (refer to "Loading a Scenario to the Device" on page 48).
- Loading an *ini* file with the ScenarioFileName parameter set to no value (i.e., ScenarioFileName = "").

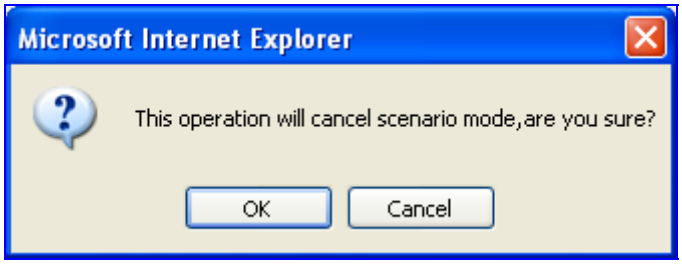
3.1.8.7 Exiting Scenario Mode

When you want to close the Scenario mode after using it for device configuration, follow the procedure below:

➢ **To close the Scenario mode:**

1. Simply click any tab (besides the **Scenarios** tab) on the Navigation bar, or click the **Cancel Scenarios** button located at the bottom of the Navigation tree; a message box appears, requesting you to confirm exiting Scenario mode, as shown below.

Figure 3-21: Confirmation Message Box for Exiting Scenario Mode



2. Click **OK** to exit.

3.1.9 Creating a Login Welcome Message

You can create a Welcome message box (alert message) that appears after each successful login to the device's Web interface. The *ini* file table parameter WelcomeMessage allows you to create the Welcome message. Up to 20 lines of character strings can be defined for the message. If this parameter is not configured, no Welcome message box is displayed after login.

An example of a Welcome message is shown in the figure below:

Figure 3-22: User-Defined Web Welcome Message after Login

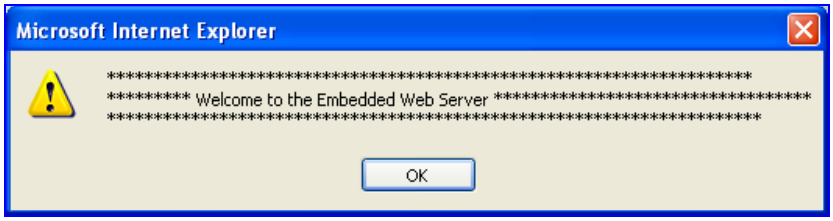


Table 3-2: *ini* File Parameter for Welcome Login Message

Parameter	Description
[WelcomeMessage]	Defines the Welcome message that appears after a successful login to the Web interface. The format of this parameter is as follows: [WelcomeMessage] FORMAT WelcomeMessage_Index = WelcomeMessage_Text; [\WelcomeMessage] For Example: [WelcomeMessage] FORMAT WelcomeMessage_Index = WelcomeMessage_Text; WelcomeMessage 1 = "**********"; WelcomeMessage 2 = "********** This is a Welcome message **"; WelcomeMessage 3 = "**********"; [\WelcomeMessage] **Note:** Each index represents a line of text in the Welcome message box. Up to 20 indices can be defined.

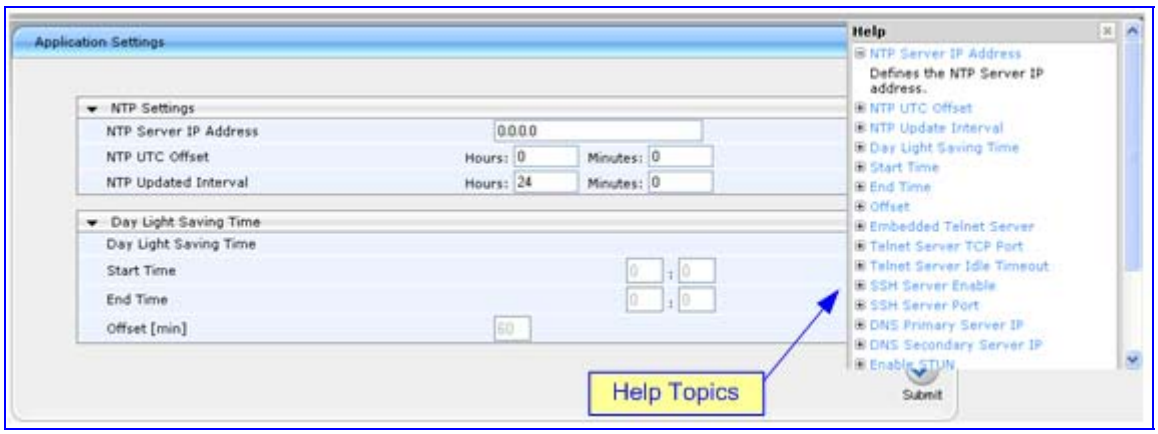
3.1.10 Getting Help

The Web interface provides you with context-sensitive Online Help. The Online Help provides you with brief descriptions of most of the parameters you'll need to successfully configure the device. The Online Help provides descriptions of parameters pertaining to the currently opened page.

➢ **To view the Help topic for a currently opened page:**

1. Using the Navigation tree, open the required page for which you want Help.
2. On the toolbar, click the **Help** button; the Help topic pertaining to the opened page appears, as shown below:

Figure 3-23: Help Topic for Current Page



3. To view a description of a parameter, click the **plus** ⊞ sign to expand the parameter. To collapse the description, click the **minus** ⊟ sign.
4. To close the Help topic, click the **close** button located on the top-right corner of the Help topic window.

Note: Instead of clicking the **Help** button for each page you open, you can open it once for a page, and then simply leave it open. Each time you open a different page, the Help topic pertaining to that page is automatically displayed.

3.1.11 Logging Off the Web Interface

You can log off the Web interface and re-access it with a different user account. For detailed information on the Web User Accounts, refer to User Accounts.

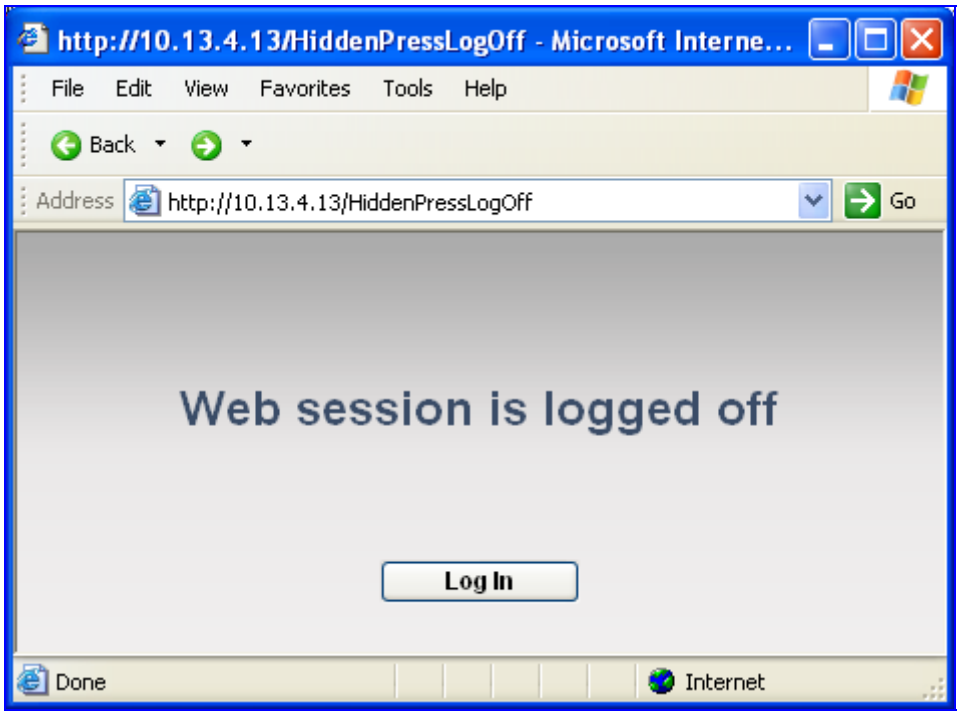
➢ **To log off the Web interface:**

1. On the toolbar, click the **Log Off** button; the 'Log Off' confirmation message box appears:

Figure 3-24: Log Off Confirmation Box

2. Click **OK**; the Web session is logged off and the **Log In** button appears.

Figure 3-25: Web Session Logged Off



To log in again, simply click the **Log In** button, and then in the 'Enter Network Password' dialog box, enter your user name and password (refer to "Accessing the Web Interface" on page 28).

3.2 Using the Home Page

The 'Home' page provides you with a graphical display of the device's front panel, displaying color-coded status icons for monitoring the functioning of the device. The 'Home' page also displays general device information (in the 'General Information' pane) such as the device's IP address and firmware version.

By default, the 'Home' page is displayed when you access the device's Web interface.

➢ **To access the Home page:**

- On the toolbar, click the **Home** icon; the 'Home' page is displayed.

Figure 3-26: Mediant 600 Home Page

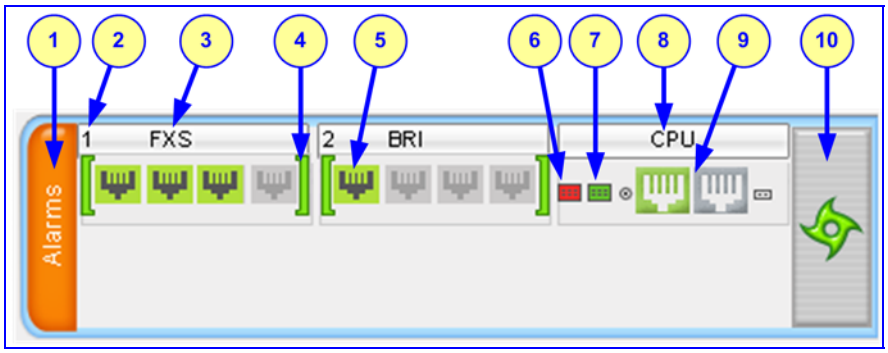
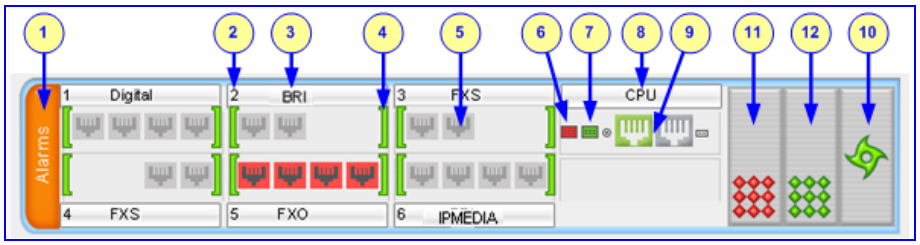


Figure 3-27: Mediant 1000 Home Page



Note: The displayed number and type of modules depends on the device's hardware configuration.

The table below describes the areas of the 'Home' page.

Table 3-3: Description of the Areas of the Home Page

Item #	Description
1	Displays the highest severity of an active alarm raised (if any) by the device: ▪ Green = No alarms ▪ Red = Critical alarm ▪ Orange = Major alarm ▪ Yellow = Minor alarm To view a list of active alarms in the 'Active Alarms' page (refer to Viewing Active Alarms on page 213), click the Alarms area.

<table>
<tr><th>Item #</th><th>Description</th></tr>
<tr><td>2</td><td>Module slot number (1 to 26).</td></tr>
<tr><td>3</td><td>Module type: FXS, FXO, DIGITAL (i.e., E1/T1), BRI, IPMEDIA.</td></tr>
<tr><td>4</td><td>Module status icon:
▪ (green): Module has been inserted or is correctly configured.
▪ (gray): Module was removed. 'Reserved' is displayed alongside the module's name.
▪ (red): Module failure. 'Failure' is displayed instead of the module's name.</td></tr>
<tr><td>5</td><td>Port (trunk or channel) status icon (refer to Viewing the Trunks' and Channels on page 58).
<table>
<tr><th>Icon</th><th>Trunk Description (Digital Module)</th><th>Channel Description (Analog Module)</th></tr>
<tr><td>(grey)</td><td>Disable: Trunk not configured (not in use)</td><td>Inactive: Channel is currently on-hook</td></tr>
<tr><td>(green)</td><td>Active - OK: Trunk synchronized</td><td>Call Connected: Active RTP stream</td></tr>
<tr><td>(yellow)</td><td>RAI Alarm: Remote Alarm Indication (RAI), also known as the Yellow Alarm</td><td>-</td></tr>
<tr><td>(red)</td><td>LOS / LOF Alarm: Loss due to LOS (Loss of Signal) or LOF (Loss of Frame)</td><td>Not Connected: No analog line is connected to this port (FXO only)</td></tr>
<tr><td>(blue)</td><td>AIS Alarm: Alarm Indication Signal (AIS), also known as the Blue Alarm</td><td>Handset Offhook: Channel is off-hook, but there is no active RTP session</td></tr>
<tr><td>(orange)</td><td>D-Channel Alarm: D-channel alarm</td><td>-</td></tr>
</table></td></tr>
<tr><td>6</td><td>Dry Contact (normally open) status icon
▪ (green): Dry Contact is open (normal).
▪ (red): Dry contact is closed.</td></tr>
<tr><td>7</td><td>Dry Contact (normally closed) status icon:
▪ (green): Dry Contact is closed (normal).
▪ (red): Dry contact is open.</td></tr>
<tr><td>8</td><td>CPU module.</td></tr>
<tr><td>9</td><td>Ethernet LAN port status icons:
▪ (green): Ethernet link is working.
▪ (gray): Ethernet link is not configured.
You can also view detailed Ethernet port information in the 'Ethernet Port Information' page (refer to Viewing Ethernet Port Information on page 210), by clicking the icon.</td></tr>
</table>

Item #	Description
10	Fan tray unit status icon: ▪ (green): Fan tray operating. ▪ (red): Fan tray failure.
11	Power Supply Unit 1 status icon (applicable only to Mediant 1000): ▪ (green): Power supply is operating. ▪ (red): Power supply failure or no power supply unit installed.
12	Power Supply Unit 2 status indicator (applicable only to Mediant 1000). Refer to Item #10 for an explanation.

3.2.1 Assigning a Port Name

The 'Home' page allows you to assign an arbitrary name or a brief description to each port. This description appears as a tooltip when you move your mouse over the port.

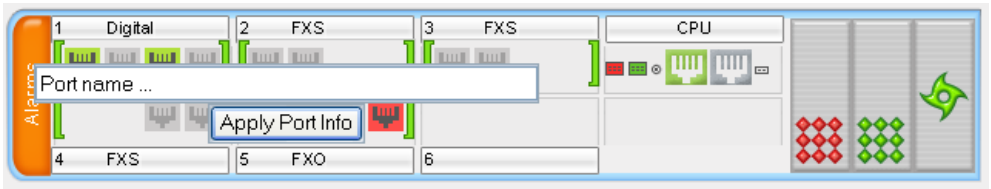
➢ **To add a port description:**

1. Click the required port icon; a shortcut menu appears, as shown below:

Figure 3-28: Shortcut Menu (Example, Mediant 1000)

2. From the shortcut menu, choose **Update Port Info**; a text box appears.

Figure 3-29: Text Box (Example, Mediant 1000)



3. Type a brief description for the port, and then click **Apply Port Info**.

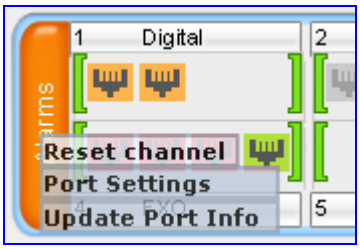
3.2.2 Resetting an Analog Channel

The 'Home' page allows you to inactivate (*reset*) an FXO or FXS analog channel. This is sometimes useful, for example, when the device (FXO) is connected to a PBX and the communication between the two can't be disconnected (e.g., when using reverse polarity).

➢ **To reset a channel:**

- Click the required **FXS** or **FXO** port icon, and then from the shortcut menu, choose **Reset Channel**; the channel is changed to inactive (i.e., the port icon is displayed in grey).

Figure 3-30: Shortcut Menu (Example, Mediant 1000)



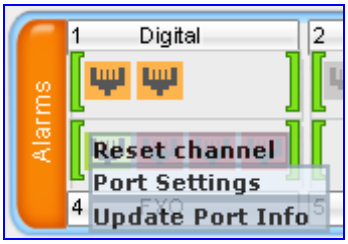
3.2.3 Viewing Analog Port Information

The 'Home' page allows you to view detailed information on a specific FXS or FXO analog port such as RTP/RTCP and voice settings.

➢ **To view detailed port information:**

1. Click the port for which you want to view port settings; the shortcut menu appears.

Figure 3-31: Selecting Port Settings from Shortcut Menu



2. From the shortcut menu, click **Port Settings**; the 'Basic Channel Information' screen appears.

Figure 3-32: Basic Information Screen

SIP ◆ Basic ◆ RTP/RTCP ◆ Voice Settings

Channel Identifier:	6
Status:	Active
Call ID:	0
Endpoint ID:	
Call Duration [sec]:	0
Call Type:	Voice
Call Destination:	0.0.0.0
Coder:	G711Alaw_64

3. To view RTP/RTCP or voice settings, click the relevant button.

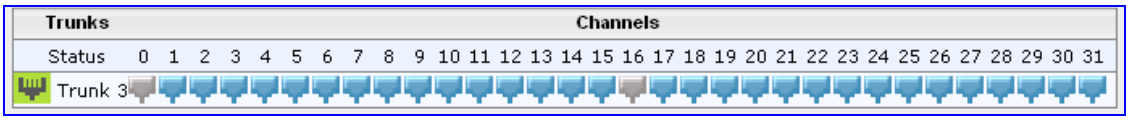
3.2.4 Viewing Trunks' Channels

The 'Home' page allows you to drill-down to view a detailed status of the channels pertaining to a trunk In addition, you can also view the trunk's configuration.

➢ **To view a detailed status of a trunk's channels:**

1. In the Home page, click the desired trunk of whose status you want to view; a shortcut menu appears.
2. From the shortcut menu, choose **Port Settings**; the 'Trunks & Channels Status' page pertaining to the specific trunk appears:

Figure 3-33: Trunks and Channels Status Screen



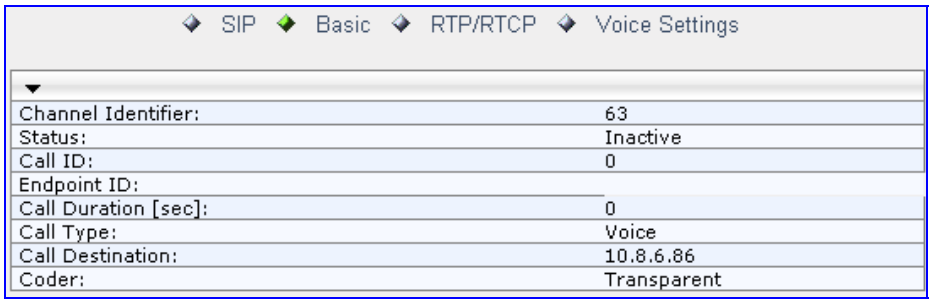
The color-coding for the status of the trunk's channels status is described in the table below:

Table 3-4: Color-Coding Status for Trunk's Channels

Icon	Color	Label	Description
	Light blue	**Inactive**	Configured, but currently no call
	Green	**Active**	Call in progress (RTP traffic)
	Purple	**SS7**	Configured for SS7 (Currently not supported)
	Grey	**Non Voice**	Not configured
	Blue	**ISDN Signaling**	Configured as a D-channel
	Yellow	**CAS Blocked**	-

3. To view the configuration settings of the trunk and / or to modify the trunk's settings, in the 'Trunks & Channels Status' page, click the **Trunk** icon, and then from the shortcut menu, choose **Port Settings**; The 'Trunk Settings' page appears. (For detailed information on configuring the trunk in this page, refer to "Configuring the Trunk Settings" on page 80.)
4. To view information of a specific trunk's channel, in the 'Trunks & Channels Status' page, click the required **Channel** icon; the 'Basic Channel Information' page appears:

Figure 3-34: Basic Channel Information Page



5. Click the buttons located above the 'Basic Channel Information' page to view additional parameters.

3.2.5 Replacing Modules

To replace the device's modules, you must use the Web interface in combination with physical removal and insertion of the modules. In other words, when you replace a module, you first need to 'software-remove' it, then extract it physically from the chassis and insert a new module, and then 'software-insert' it using the Web interface. The software removal and insertion is performed in the 'Home' page.

Warnings:

- A module must be replaced with the same type of module and in the same module slot number. For example, a module with two digital spans in Slot 1 must be replaced with a module with two digital spans in Slot 1.
- When only one module is available, removal of the module causes the device to reset.
- Before inserting a module into a previously empty slot, you must power down the device.

Note: This section is applicable only to Mediant 1000.

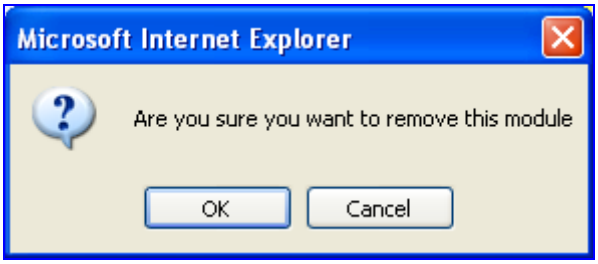
➢ **To replace a module:**

1. Remove the module by performing the following:
   a. In the 'Home' page, click the title of the module that you want to replace; the **Remove Module** button appears:

Figure 3-35: Remove Module Button Appears after Clicking Module Name

   b. Click the **Remove Module** button; a message box appears requesting you to confirm module removal:

Figure 3-36: Module Removal Confirmation Message Box



c. Click **OK** to confirm removal; after a few seconds, the module is software-removed, the module status icon turns to grey, and the name of the module is suffixed with the word 'Reserved':

Figure 3-37: Removed Module

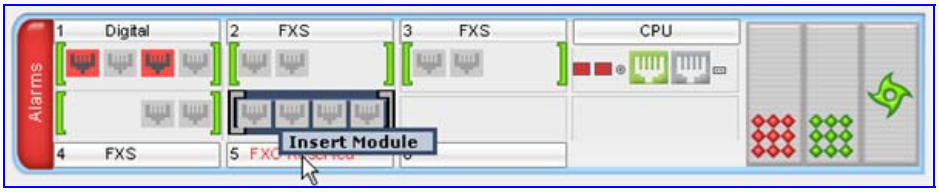
d. Physically remove the module (refer to the device's *Installation Manual*).

2. Insert the replaced module, by performing the following:

a. Physically insert the replaced module (refer to the device's *Installation Manual*) into the same slot in which the previous module resided.

b. In the 'Home' page, click the title of the module ('<module type> Reserved') that you want to replace; the **Insert Module** button appears:

Figure 3-38: Insert Module Button after Clicking Module's Name



c. Click the **Insert Module** button; a message appears informing you that this may take a few seconds. When the message disappears, the module is inserted, which is indicated by the disappearance of the word 'Reserved' from the module's name.

3.3 Configuration Tab

The **Configuration** tab on the Navigation bar displays menus in the Navigation tree related to device configuration. These menus include the following:

- **Network Settings** (refer to "Network Settings" on page 60)
- **Media Settings** (refer to "Media Settings" on page 71)
- **PSTN Settings** (refer to “PSTN Settings” on page 77)
- **Security Settings** (refer to "Security Settings" on page 82)
- **Protocol Configuration** (refer to "Protocol Configuration" on page 99)
- **TDM Configuration** (refer to “Configuring TDM Bus Settings” on page 184)
- **Advanced Applications** (refer to "Advanced Applications" on page 185)

3.3.1 Network Settings

The **Network Settings** menu allows you to configure various networking parameters. This menu includes the following items:

- IP Settings (refer to "Configuring the Multiple Interface Table" on page 61)
- Application Settings (refer to "Configuring the Application Settings" on page 65)

- IP Routing Table (refer to "Configuring the IP Routing Table" on page 68)
- QoS Settings (refer to "Configuring the QoS Settings" on page 70)

3.3.1.1 Configuring the Multiple Interface Table

The 'Multiple Interface Table' page allows you to configure up to 16 logical network interfaces, each with its own IP address, unique VLAN ID (if enabled), interface name, and application type permitted on the interface:

- Control
- Media
- Operations, Administration, Maintenance and Provisioning (OAMP)

This page also provides VLAN-related parameters for enabling VLANs and for defining the 'Native' VLAN ID (VLAN ID to which incoming, untagged packets are assigned). For assigning VLAN priorities and Differentiated Services (DiffServ) for the supported Class of Service (CoS), refer to "Configuring the QoS Settings" on page 70.

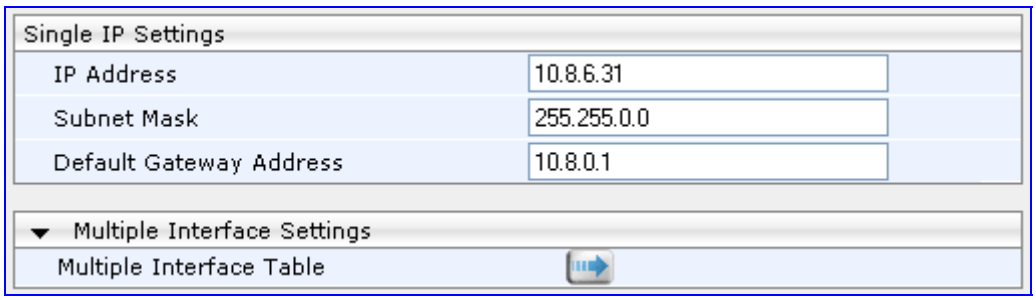
Notes:

- Only eight media (RTP) IP address interfaces can be implemented in call routing. These interfaces are assigned to Media Realms in the 'SIP Media Realm' table (refer to “Configuring Media Realms” on page 99).
- Once you access the 'Multiple Interface Table' page, the 'IP Settings' page is no longer available.
- For a detailed description with examples for configuring multiple network interfaces, refer to “Network Configuration” on page 608).
- You can view all configured IP interfaces that are currently active in the 'IP Active Interfaces' page (refer to "Viewing Active IP Interfaces" on page 211).
- When adding more than one interface to the table, ensure that you enable VLANs using the 'VLAN Mode' (VlANMode) parameter.
- When booting using BootP/DHCP protocols (refer to the Product Reference Manual), an IP address is obtained from the server. This address is used as the OAMP address for this session, overriding the IP address you configured in the 'Multiple Interface Table' page. The address specified in this table takes effect only after you save the configuration to the device's flash memory. This enables the device to use a temporary IP address for initial management and configuration, while retaining the address (defined in this table) for deployment.
- For an explanation on configuring tables in the Web interface, refer to "Working with Tables" on page 39.
- You can also configure this table using the *ini* file table parameter InterfaceTable (refer to "Networking Parameters" on page 247).

➢ **To configure the multiple IP interface table:**

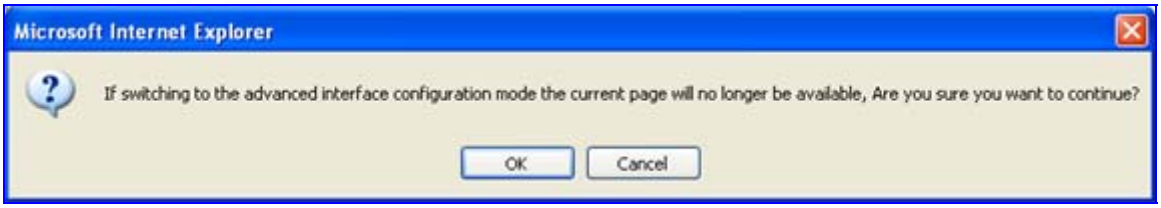
1. Open the 'IP Settings' page (**Configuration** tab > **Network Settings** menu > **IP Settings**).

Figure 3-39: IP Settings Page



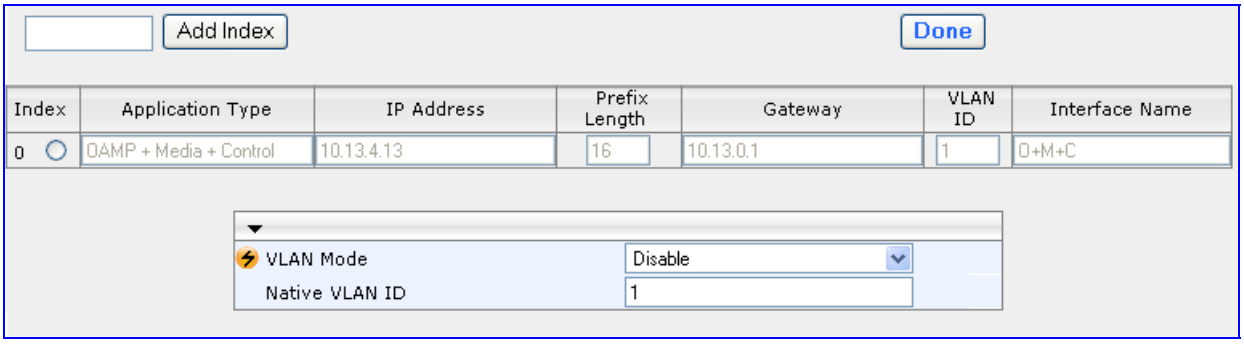
2. Under the 'Multiple Interface Settings' group, click the Multiple Interface Table button; a confirmation message box appears:

Figure 3-40: Confirmation Message for Accessing the Multiple Interface Table


Microsoft Internet Explorer
If switching to the advanced interface configuration mode the current page will no longer be available, Are you sure you want to continue?
OK
Cancel

3. Click **OK** to confirm; the 'Multiple Interface Table' page appears:

Figure 3-41: Multiple Interface Table Page



4. In the 'Add Index' field, enter the desired index number for the new interface, and then click **Add**; the index row is added to the table.
5. Configure the interface according to the table below.
6. Click the **Apply** button; the interface is added to the table and the **Done** button appears.
7. Click **Done** to validate the interface. If the interface is not a valid (e.g., if it overlaps with another interface in the table or it does not adhere to the other rules for adding interfaces), a message is displayed to inform you and you must redefine your interfaces accordingly.
8. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

Table 3-5: Multiple Interface Table Parameters Description

Parameter	Description
Table parameters	
Index	Index of each interface. The range is 0 to 15. **Note:** Each interface index must be unique.
Web: Application Type EMS: Application Types **[InterfaceTable_ApplicationTypes]**	Types of applications that are allowed on the specific interface. ▪ **[0]** OAMP = Only Operations, Administration, Maintenance and Provisioning (OAMP) applications (e.g., Web, Telnet, SSH, and SNMP) are allowed on the interface. ▪ **[1]** Media = Only Media (i.e., RTP streams of voice) is allowed on the interface. ▪ **[2]** Control = Only Call Control applications (e.g., SIP) are allowed on the interface. ▪ **[3]** OAMP + Media = Only OAMP and Media applications are allowed on the interface. ▪ **[4]** OAMP + Control = Only OAMP and Call Control applications are allowed on the interface. ▪ **[5]** Media + Control = Only Media and Call Control applications are allowed on the interface. ▪ **[6]** OAMP + Media + Control = All application types are allowed on the interface. **Notes:** ▪ A single OAMP interface (and only one) must be configured. This OAMP interface can be combined with Media and Control. ▪ At least one interface with Media and at least one interface with Control must be configured. ▪ Multiple interfaces for Media, Control, and Media and Control can be configured. ▪ At least one IPv4 interface with Control must be configured. This can be combined with OAMP and Media. ▪ At least one IPv4 interface with Media must be configured. This can be combined with OAMP and Control.
Web/EMS: IP Address **[InterfaceTable_IPAddres]**	The IPv4 IP address in dotted-decimal notation. **Notes:** ▪ Each interface must be assigned a unique IP address. ▪ When booting using BootP/DHCP protocols, an IP address is obtained from the server. This address is used as the OAMP address for the initial session, overriding the address configured using the InterfaceTable. The address configured for OAMP applications in this table becomes available when booting from flash again. This enables the device to operate with a temporary address for initial management and configuration while retaining the address to be used for deployment.

Parameter	Description
Web/EMS: Prefix Length **[InterfaceTable_PrefixLength]**	Defines the Classless Inter-Domain Routing (CIDR)-style representation of a dotted decimal subnet notation. The CIDR-style representation uses a suffix indicating the number of bits which are set in the dotted decimal format (e.g. 192.168.0.0/16 is synonymous with 192.168.0.0 and a subnet of 255.255.0.0. Defines the number of '1' bits in the subnet mask (i.e., replaces the standard dotted-decimal representation of the subnet mask for IPv4 interfaces). For example: A subnet mask of 255.0.0.0 is represented by a prefix length of 8 (i.e., 11111111 00000000 00000000 00000000), and a subnet mask of 255.255.255.252 is represented by a prefix length of 30 (i.e., 11111111 11111111 11111111 11111100). The prefix length is a Classless Inter-Domain Routing (CIDR) style presentation of a dotted-decimal subnet notation. The CIDR-style presentation is the latest method for interpretation of IP addresses. Specifically, instead of using eight-bit address blocks, it uses the variable-length subnet masking technique to allow allocation on arbitrary-length prefixes (refer to http://en.wikipedia.org/wiki/Classless_Inter-Domain_Routing for more information). For IPv4 Interfaces, the prefix length values range from 0 to 31. **Note:** Subnets of different interfaces must not overlap in any way (e.g., defining two interfaces with 10.0.0.1/8 and 10.50.10.1/24 is invalid). Each interface must have its own address space.
Web/EMS: Gateway **[InterfaceTable_Gateway]**	Defines the IP address of the default gateway used by the device. **Notes:** ▪ Only one default gateway can be defined. ▪ The default gateway must be configured on an interface that includes Media traffic. ▪ The default gateway's IP address must be in the same subnet as the interface address. ▪ Apart from the interface with the defined default gateway, for all other interfaces define this parameter to "0.0.0.0". ▪ For configuring additional routing rules for other interfaces, use the Routing table (refer to "Configuring the IP Routing Table" on page 68).
Web/EMS: VLAN ID **[InterfaceTable_VlanID]**	Defines the VLAN ID for each interface. Incoming traffic with this VLAN ID is routed to the corresponding interface, and outgoing traffic from that interface is tagged with this VLAN ID. **Notes:** ▪ The VLAN ID must be unique for each interface. ▪ VLANs are available only when booting the device from flash. When booting using BootP/DHCP protocols, VLANs are disabled to allow easier maintenance access. In this scenario, multiple network interface capabilities are not available.

Parameter	Description
Web/EMS: Interface Name **[InterfaceTable_InterfaceName]**	Defines a string (up to 16 characters) to name this interface. This name is displayed in management interfaces (Web, CLI and SNMP) for better readability (and has no functional use) as well as the 'SIP Media Realm' table (refer to "Configuring Media Realms" on page 99). **Note:** The interface name is a mandatory parameter and must be unique for each interface.
General Parameters	
VLAN Mode **[VlANMode]**	For a description of this parameter, refer to “Networking Parameters” on page 247.
Native VLAN ID **[VLANNativeVlanID]**	For a description of this parameter, refer to “Networking Parameters” on page 247.

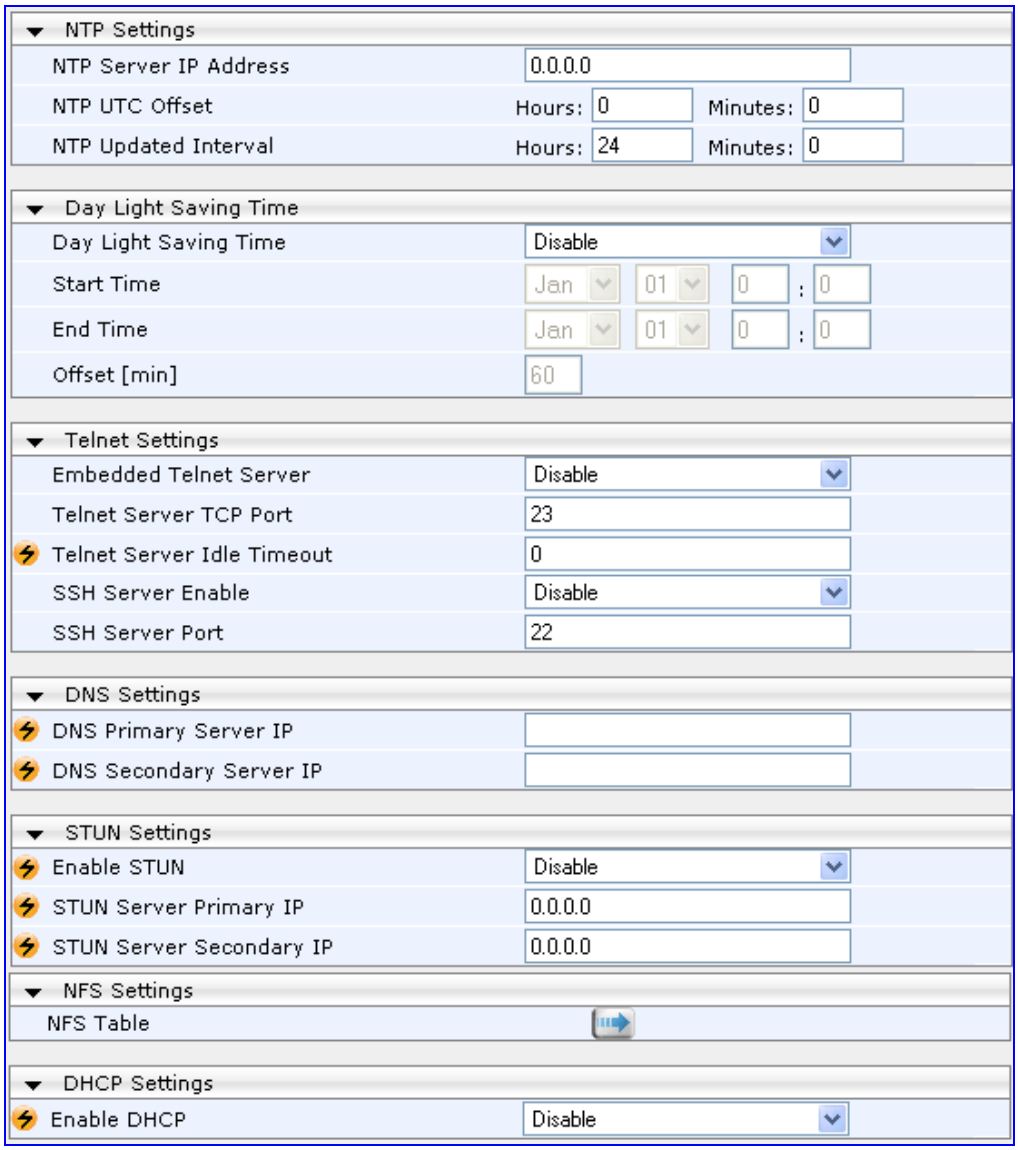
3.3.1.2 Configuring the Application Settings

The 'Application Settings' page is used for configuring various application parameters such as Network Time Protocol (NTP), daylight saving time, and Telnet. For a description of these parameters, refer to "Configuration Parameters Reference" on page 247.

➢ **To configure the Application settings:**

1. Open the 'Application Settings' page (**Configuration** tab > **Network Settings** menu > **Application Settings** page item).

Figure 3-42: Application Settings Page



2. Configure the parameters as required. For configuring NFS, under the 'NFS Settings' group, click the NFS Table button; the 'NFS Settings' page appears. For a description on configuring this page, refer to Configuring the NFS Settings on page 67.
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

3.3.1.3 Configuring the NFS Settings

Network File System (NFS) enables the device to access a remote server's shared files and directories, and to handle them as if they're located locally. You can configure up to 16 different NFS file systems. As a file system, the NFS is independent of machine types, operating systems, and network architectures. NFS is used by the device to load the *cmp*, *ini*, and auxiliary files, using the Automatic Update mechanism (refer to the *Product Reference Manual*). Note that an NFS file server can share multiple file systems. There must be a separate row for each remote file system shared by the NFS file server that needs to be accessed by the device.

➢ **To add remote NFS file systems:**

1. Open the 'Application Settings' page (refer to "Configuring the Application Settings" on page 65).
2. Under the NFS Settings group, click the **NFS Table** button; the 'NFS Settings' page appears.

Figure 3-43: NFS Settings Page

Index	Host Or IP	Root Path	NFS Version	Authentication Type	User ID	GID	Vlan Type
1	10.13.4.5	/audio_files	NFS Version 3	1	0	1	MEDIA

3. In the 'Add' field, enter the index number of the remote NFS file system, and then click **Add**; an empty entry row appears in the table.
4. Configure the NFS parameters according to the table below.
5. Click the **Apply** button; the remote NFS file system is immediately applied, which can be verified by the appearance of the 'NFS mount was successful' message in the Syslog server.
6. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

Notes:

- To avoid terminating current calls, a row must not be deleted or modified while the device is currently accessing files on that remote NFS file system.
- The combination of 'Host Or IP' and 'Root Path' must be unique for each row in the table. For example, the table must include only one row with a Host / IP of 192.168.1.1 and Root Path of /audio.
- For an explanation on configuring Web interface tables, refer to "Working with Tables" on page 39.
- You can also configure the NFS table using the *ini* file table parameter NFSServers (refer to "NFS Parameters" on page 256).

Table 3-6: NFS Settings Parameters

Parameter	Description
Index	The row index of the remote file system. The valid range is 1 to 16.
Host Or IP	The domain name or IP address of the NFS server. If a domain name is provided, a DNS server must be configured.
Root Path	Path to the root of the remote file system in the format: /**[path].** For example, '/audio'.
NFS Version	NFS version used to access the remote file system. ▪ **[2]** NFS Version 2. ▪ **[3]** NFS Version 3 (default).
Authentication Type	Authentication method used for accessing the remote file system. ▪ **[0]** Null ▪ **[1]** Unix (default)
User ID	User ID used in authentication when using Unix. The valid range is 0 to 65537. The default is 0.
Group ID	Group ID used in authentication when using Unix. The valid range is 0 to 65537. The default is 1.
VLAN Type	The VLAN type for accessing the remote file system. ▪ **[0]** OAM ▪ **[1]** MEDIA (default) **Note:** This parameter applies only if VLANs are enabled or if Multiple IPs is configured (refer to "Network Configuration" on page 608).

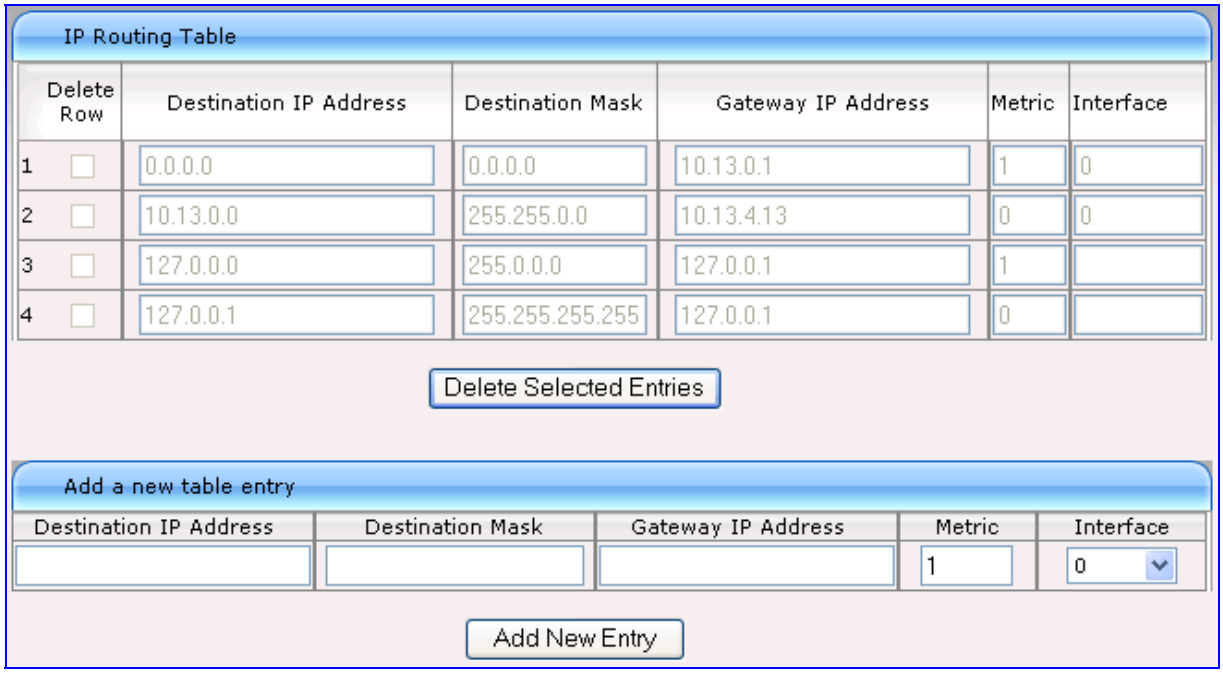
3.3.1.4 Configuring the IP Routing Table

The 'IP Routing Table' page allows you to define up to 50 static IP routing rules for the device. For example, you can define static routing rules for the OAMP and Control networks since a default gateway is supported only for the Media traffic network. Before sending an IP packet, the device searches this table for an entry that matches the requested destination host / network. If such an entry is found, the device sends the packet to the indicated router. If no explicit entry is found, the packet is sent to the default gateway (refer to "Configuring the Multiple Interface Table" on page 61).

➢ **To configure static IP routing:**

1. Open the 'IP Routing Table' page (**Configuration** tab > **Network Settings** menu > **IP Routing Table** page item).

Figure 3-44: IP Routing Table Page



2. In the 'Add a new table entry' group, add a new static routing rule according to the parameters described in the table below.
3. Click **Add New Entry**; the new routing rule is added to the IP routing table.

To delete a routing rule from the table, select the 'Delete Row' check box that corresponds to the routing rule entry, and then click **Delete Selected Entries**.

Table 3-7: IP Routing Table Description

Parameter	Description
Destination IP Address **[RoutingTableDestinationsColumn]**	Specifies the IP address of the destination host / network.
Destination Mask **[RoutingTableDestinationMasksColumn]**	Specifies the subnet mask of the destination host / network.
The address of the host / network you want to reach is determined by an AND operation that is applied to the fields 'Destination IP Address' and 'Destination Mask'. For example, to reach the network 10.8.x.x, enter 10.8.0.0 in the field 'Destination IP Address' and 255.255.0.0 in the field 'Destination Mask'. As a result of the AND operation, the value of the last two octets in the field 'Destination IP Address' is ignored. To reach a specific host, enter its IP address in the field 'Destination IP Address' and 255.255.255.255 in the field 'Destination Mask'.	
Gateway IP Address **[RoutingTableGatewaysColumn]**	The IP address of the router (next hop) to which the packets are sent if their destination matches the rules in the adjacent columns. **Note:** The Gateway address must be in the same subnet on which the address is configured on the

Parameter	Description
	'Multiple Interface Table' page (refer to "Configuring the Multiple Interface Table" on page 61).
Metric **[RoutingTableHopsCountColumn]**	The maximum number of times a packet can be forwarded (hops) between the device and destination (typically, up to 20). **Note:** This parameter must be set to a number greater than 0 for the routing rule to be valid. Routing entries with Hop Count equals 0 are local routes set automatically by the device.
Interface **[RoutingTableInterfacesColumn]**	Specifies the interface (network type) to which the routing rule is applied. ▪ **[0]** = OAMP (default). ▪ **[1]** = Media. ▪ **[2]** = Control. For detailed information on the network types, refer to "Configuring the Multiple Interface Table" on page 61.

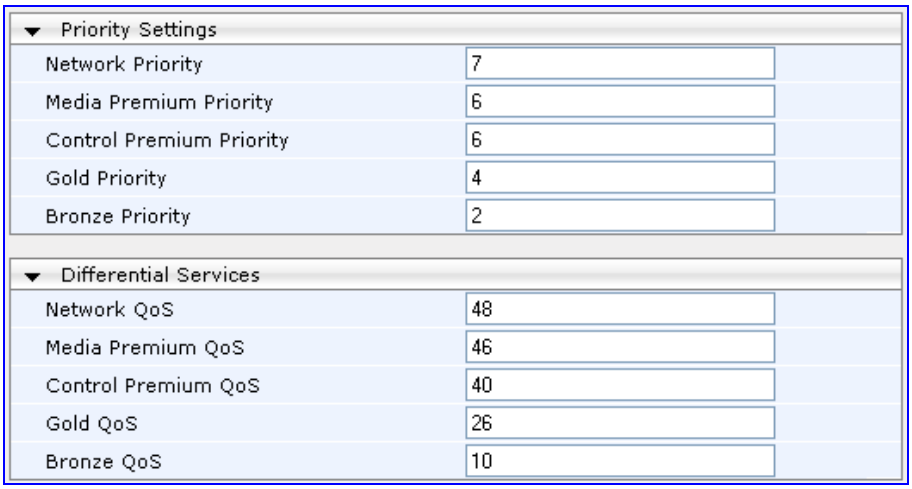
3.3.1.5 Configuring the QoS Settings

The 'QoS Settings' page is used for configuring the Quality of Service (QoS) parameters. This page allows you to assign VLAN priorities (IEEE 802.1p) and Differentiated Services (DiffServ) for the supported Class of Service (CoS). For a detailed description of the parameters appearing on this page, refer to "Networking Parameters" on page 247. For detailed information on IP QoS using DiffServ, refer to "IP QoS via Differentiated Services (DiffServ)" on page 608.

➢ **To configure QoS:**

1. Open the 'QoS Settings' page (**Configuration** tab > **Network Settings** menu > **QoS Settings** page item).

Figure 3-45: QoS Settings Page



2. Configure the QoS parameters as required.
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

3.3.2 Media Settings

The **Media Settings** menu allows you to configure the device's channel parameters. This menu contains the following items:

- Voice Settings (refer to "Configuring the Voice Settings" on page 71)
- Fax/Modem/CID Settings (refer to “Configuring the Fax/Modem/CID Settings” on page 73)
- RTP/RTCP Settings (refer to "Configuring the RTP/RTCP Settings" on page 74)
- IP media Settings (refer to “Configuring the IP Media Settings” on page 75)
- General Media Settings (refer to "Configuring the General Media Settings" on page 75)
- Analog Settings (refer to “Configuring the Analog Settings” on page 76)
- Media Security (refer to "Configuring Media Security" on page 77)

Notes:

- Channel parameters can be modified on-the-fly. Changes take effect from the next call.
- Some channel parameters can be configured per channel or call routing, using profiles (refer to “Coders and Profile Definitions” on page 127).

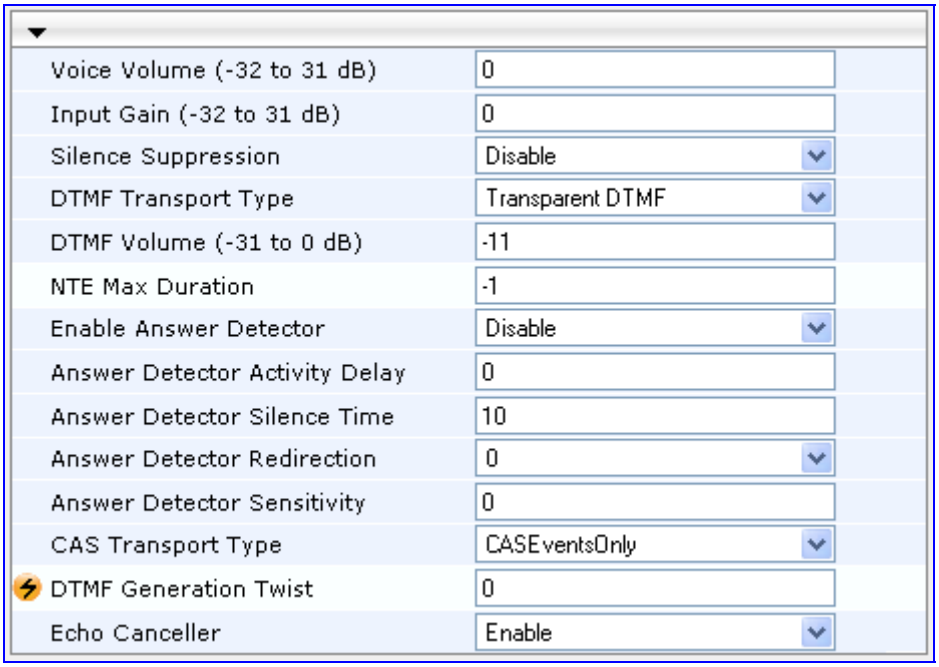
3.3.2.1 Configuring the Voice Settings

The 'Voice Settings' page is used for configuring various voice parameters such as voice volume, silence suppression, and DTMF transport type. For a detailed description of the parameters appearing on this page, refer to "Configuration Parameters Reference" on page 247.

➢ **To configure the Voice parameters:**

1. Open the 'Voice Settings' page (**Configuration** tab > **Media Settings** menu > **Voice Settings** page item).

Figure 3-46: Voice Settings Page



2. Configure the Voice parameters as required.
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

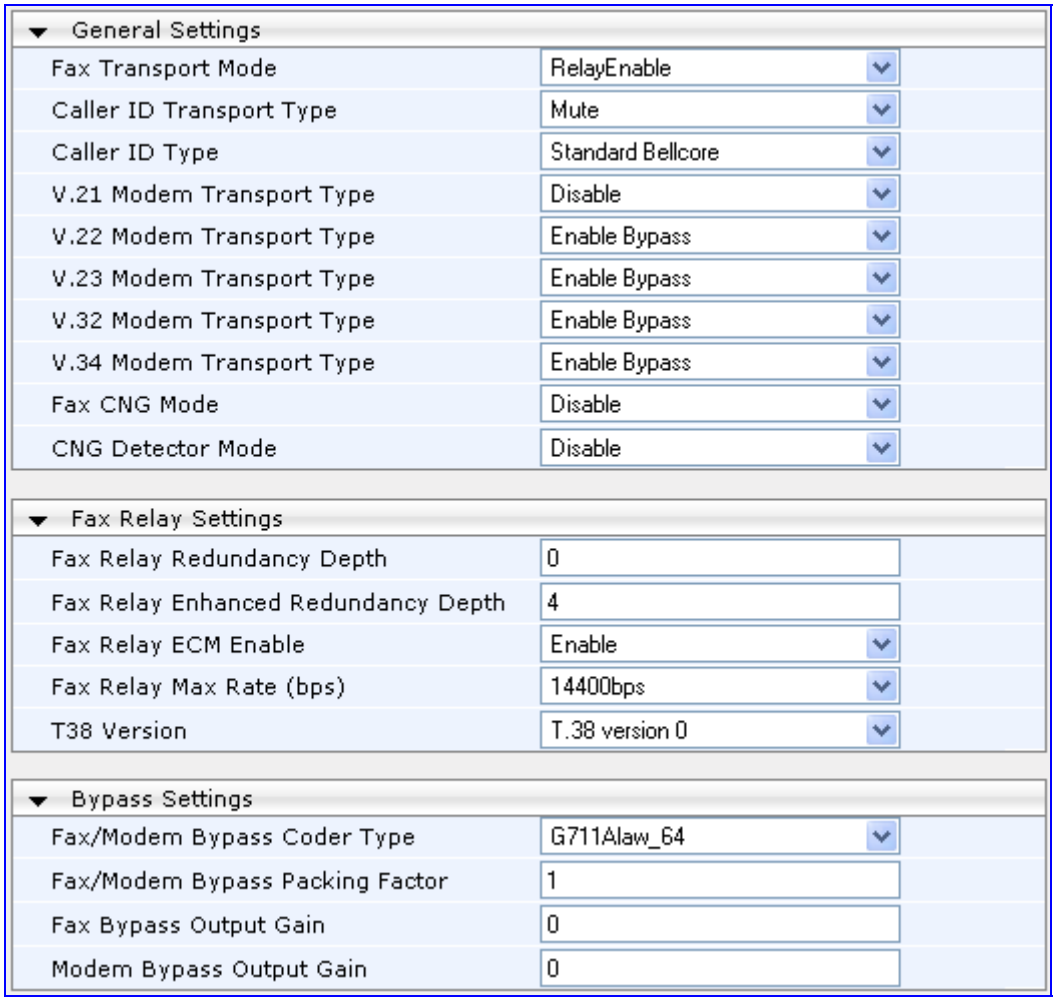
3.3.2.2 Configuring the Fax/Modem/CID Settings

The 'Fax/Modem/CID Settings' page is used for configuring fax, modem, and Caller ID (CID) parameters. For a detailed description of the parameters appearing on this page, refer to "Configuration Parameters Reference" on page 247.

➢ **To configure the fax, modem, and CID parameters:**

1. Open the 'Fax/Modem/CID Settings' page (**Configuration** tab > **Media Settings** menu > **Fax/Modem/CID Settings** page item).

Figure 3-47: Fax/Modem/CID Settings Page



2. Configure the parameters as required.
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

Note: Some SIP parameters override these fax and modem parameters (refer to the parameter IsFaxUsed, and V.152 parameters in Section "V.152 Support" on page 563).

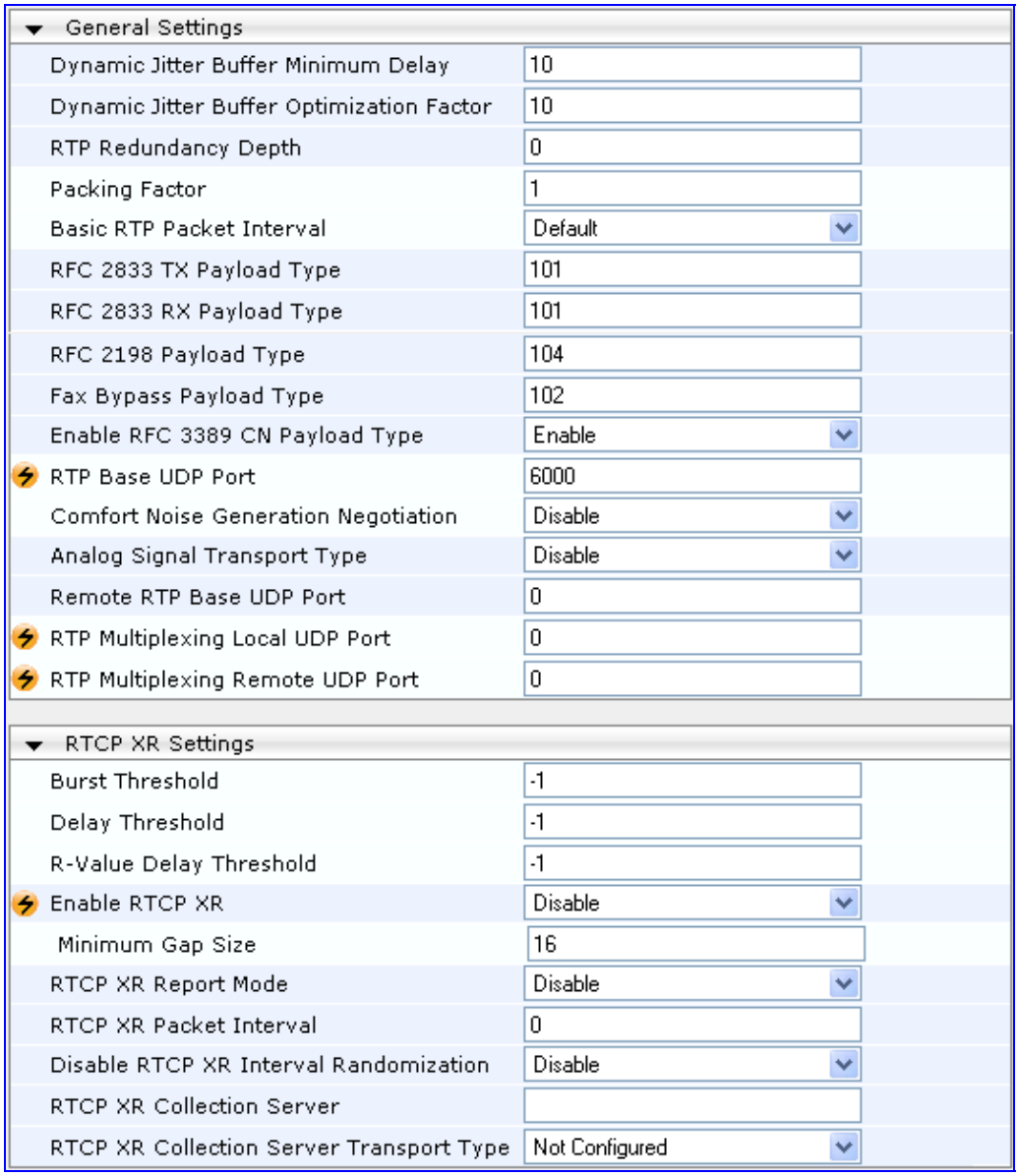
3.3.2.3 Configuring the RTP/RTCP Settings

The 'RTP/RTCP Settings' page allows you to configure the Real-Time Transport Protocol (RTP) and Real-Time Transport (RTP) Control Protocol (RTCP) parameters. For a detailed description of the parameters appearing on this page, refer to "Configuration Parameters Reference" on page 247.

➢ **To configure the RTP/RTCP parameters:**

1. Open the 'RTP/RTCP Settings' page (**Configuration** tab > **Media Settings** menu > **RTP / RTCP Settings** page item).

Figure 3-48: RTP/RTCP Settings Page



2. Configure the parameters as required.
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

3.3.2.4 Configuring the IP Media Settings

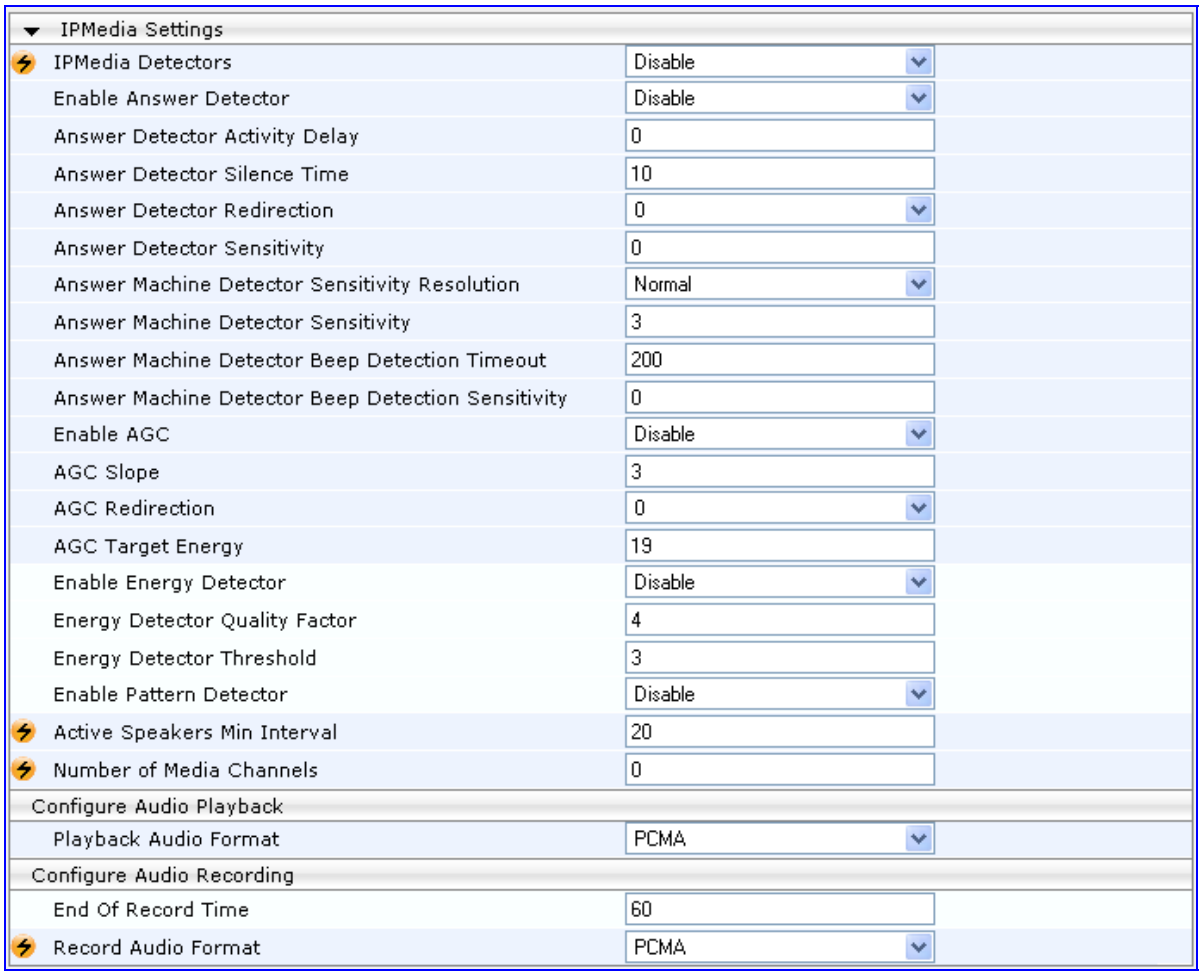
The 'IPMedia Settings' page allows you to configure the IP media parameters.

For a detailed description of the parameters appearing on this page, refer to "Configuration Parameters Reference" on page 247.

➢ **To configure the IP media parameters:**

1. Open the 'IPMedia Settings' page (**Configuration** tab > **Media Settings** menu > **IPMedia Settings** page item).

Figure 3-49: IPMedia Settings Page



2. Configure the parameters as required.
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

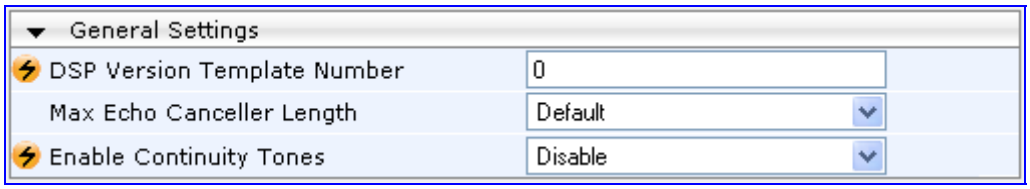
3.3.2.5 Configuring the General Media Settings

The 'General Media Settings' page allows you to configure various media parameters. For a detailed description of the parameters appearing on this page, refer to "Configuration Parameters Reference" on page 247.

➢ **To configure general media parameters:**

1. Open the 'General Media Settings' page (**Configuration** tab > **Media Settings** menu > **General Media Settings** page item).

Figure 3-50: General Media Page



2. Configure the parameters as required.
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

3.3.2.6 Configuring the Analog Settings

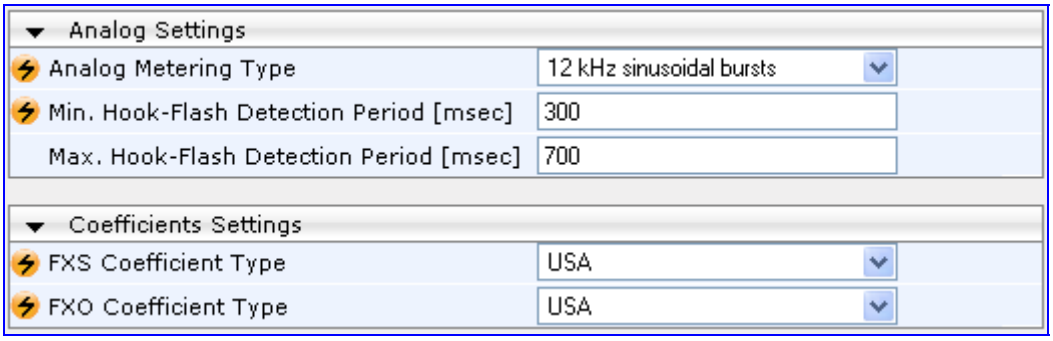
The 'Analog Settings' page allows you to configure various analog parameters. For a detailed description of the parameters appearing on this page, refer to "Configuration Parameters Reference" on page 247.

This page also selects the type (USA or Europe) of FXS and/or FXO coefficient information. The FXS coefficient contains the analog telephony interface characteristics such as DC and AC impedance, feeding current, and ringing voltage.

➢ **To configure the analog parameters:**

1. Open the 'Analog Settings' page (**Configuration** tab > **Media Settings** menu > **Analog Settings** page item).

Figure 3-51: Analog Settings Page



2. Configure the parameters as required.
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

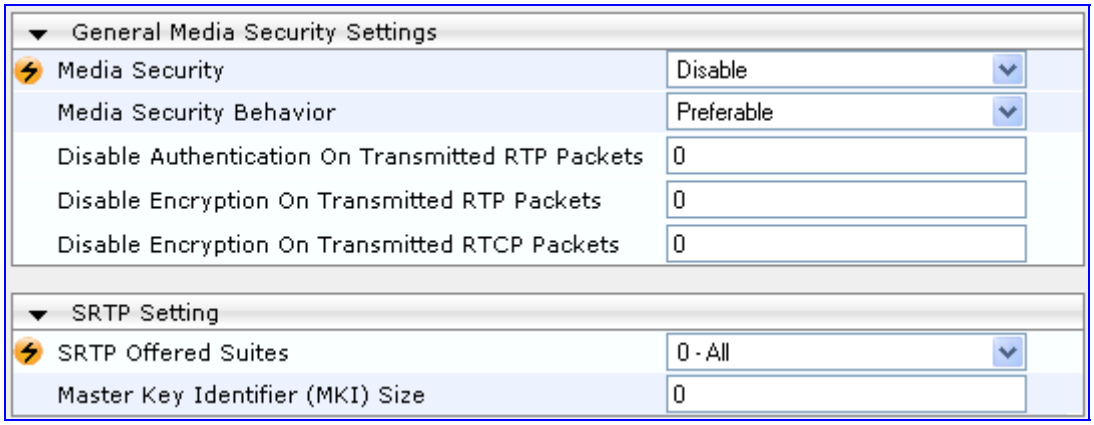
3.3.2.7 Configuring Media Security

The 'Media Security' page allows you to configure media security. For a detailed description of the parameters appearing on this page, refer to "Configuration Parameters Reference" on page 247.

➢ **To configure media security:**

1. Open the 'Media Security' page (**Configuration** tab > **Media Settings** menu > **Media Security** page item).

Figure 3-52: Media Security Page



2. Configure the parameters as required.
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

3.3.3 PSTN Settings

The **PSTN Settings** menu allows you to configure various PSTN settings and includes the following page items:

- CAS State Machines (refer to "Configuring the CAS State Machines" on page 77)
- Trunk Settings (refer to "Configuring the Trunk Settings" on page 80)

3.3.3.1 Configuring the CAS State Machines

The 'CAS State Machine' page allows you to modify various timers and other basic parameters to define the initialization of the CAS state machine without changing the state machine itself (no compilation is required). The change doesn't affect the state machine itself, but rather the configuration.

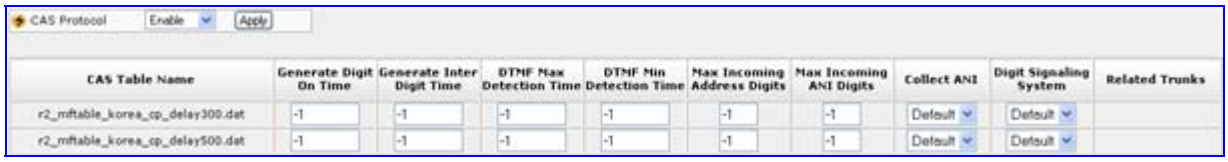
The CAS table used can be chosen in two ways (using the parameter CasChannelIndex):

- Single CAS table per Trunk.
- Different CAS table per group of B-Channels in a trunk.

➢ **To modify the CAS state machine parameters:**

1. Open the 'CAS State Machine' page (**Configuration** tab > **PSTN Settings** menu > **CAS State Machines** page item).

Figure 3-53: CAS State Machine Page



2. Ensure that the trunk is inactive. The trunk number displayed in the 'Related Trunks' field must be green. If it is red (indicating that the trunk is active), click the trunk number to open the 'Trunk Settings' page (refer to "Configuring the Trunk Settings" on page 80), select the required Trunk number icon, and then click **Stop Trunk**.
3. In the 'CAS State Machine' page, modify the required parameters according to the table below.
4. Once you have completed the configuration, activate the trunk if required in the 'Trunk Settings' page, by clicking the trunk number in the 'Related Trunks' field, and in the 'Trunk Settings' page, select the required Trunk number icon, and then click **Apply Trunk Settings**.
5. Click **Submit**.
6. Reset the device (refer to "Resetting the Device" on page 197).

Notes:

- It's strongly recommended that you don't modify the default values unless you fully understand the implications of the changes and know the default values. Every change affects the configuration of the state machine parameters and the call process related to the trunk you are using with this state machine.
- You can modify CAS state machine parameters only if the following conditions are met:
  1) Trunks are inactive (stopped), i.e., the 'Related Trunks' field displays the trunk number in green.
  2) State machine is not in use or is in reset, or when it is not related to any trunk. If it is related to a trunk, you must delete the trunk or de-activate (*Stop*) the trunk.
- Field values displaying '-1' indicate CAS default values. In other words, CAS state machine values are used.
- The modification of the CAS state machine occurs at the CAS application initialization only for non-default values (-1).
- For a detailed description of the CAS Protocol table, refer to the *Product Reference Manual*.

Table 3-8: CAS State Machine Parameters Description

Parameter	Description
Generate Digit On Time **[CasStateMachineGenerateDigitOnTime]**	Generates digit on-time (in msec). The value must be a positive value. The default value is -1 (use value from CAS state machine).
Generate Inter Digit Time **[CasStateMachineGenerateInterDigitTime]**	Generates digit off-time (in msec). The value must be a positive value. The default value is -1 (use value from CAS state machine).
DTMF Max Detection Time **[CasStateMachineDTMFMaxOnDetectionTime]**	Detects digit maximum on time (according to DSP detection information event) in msec units. The value must be a positive value. The default value is -1 (use value from CAS state machine).
DTMF Min Detection Time **[CasStateMachineDTMFMinOnDetectionTime]**	Detects digit minimum on time (according to DSP detection information event) in msec units. The digit time length must be longer than this value to receive a detection. Any number may be used, but the value must be less than CasStateMachineDTMFMaxOnDetectionTime. The value must be a positive value. The default value is -1 (use value from CAS state machine).
MAX Incoming Address Digits **[CasStateMachineMaxNumOfIncomingAddressDigits]**	Defines the limitation for the maximum address digits that need to be collected. After reaching this number of digits, the collection of address digits is stopped. The value must be an integer. The default value is -1 (use value from CAS state machine).
MAX Incoming ANI Digits **[CasStateMachineMaxNumOfIncomingANIDigits]**	Defines the limitation for the maximum ANI digits that need to be collected. After reaching this number of digits, the collection of ANI digits is stopped. The value must be an integer. The default value is -1 (use value from CAS state machine).
Collet ANI **[CasStateMachineCollectANI]**	In some cases, when the state machine handles the ANI collection (not related to MFCR2), you can control the state machine to collect ANI or discard ANI. ▪ **[0]** No = Don't collect ANI. ▪ **[1]** Yes = Collect ANI. ▪ **[-1]** Default = Default value - use value from CAS state machine.
Digit Signaling System **[CasStateMachineDigitSignalingSystem]**	Defines which Signaling System to use in both directions (detection\generation). ▪ **[0]** DTMF = Uses DTMF signaling. ▪ **[1]** MF = Uses MF signaling (default). ▪ **[-1]** Default = Default value - use value from CAS state machine.

3.3.3.2 Configuring the Trunk Settings

The 'Trunk Settings' page allows you to configure the device's trunks. This includes selecting the PSTN protocol and configuring related parameters.

Some parameters can be configured when the trunk is in service, while others require you to take the trunk out of service (by clicking the **Stop** button). Once you have "stopped" a trunk, all calls are dropped and no new calls can be made on that trunk.

You can also deactivate a trunk (by clicking the **Deactivate** button) for maintenance. Deactivation temporarily disconnects (logically) the trunk from the PSTN network. Upon trunk deactivation, the device generates an AIS alarm on that trunk to the far-end (as a result, an RAI alarm signal may be received by the device). A subsequent trunk activation (by clicking the **Activate** button), reconnects the trunk to the PSTN network and clears the AIS alarm. Trunk deactivation is typically used for maintenance such as checking the trunk's physical integrity.

For a description of the trunk parameters, refer to "PSTN Parameters" on page 375.

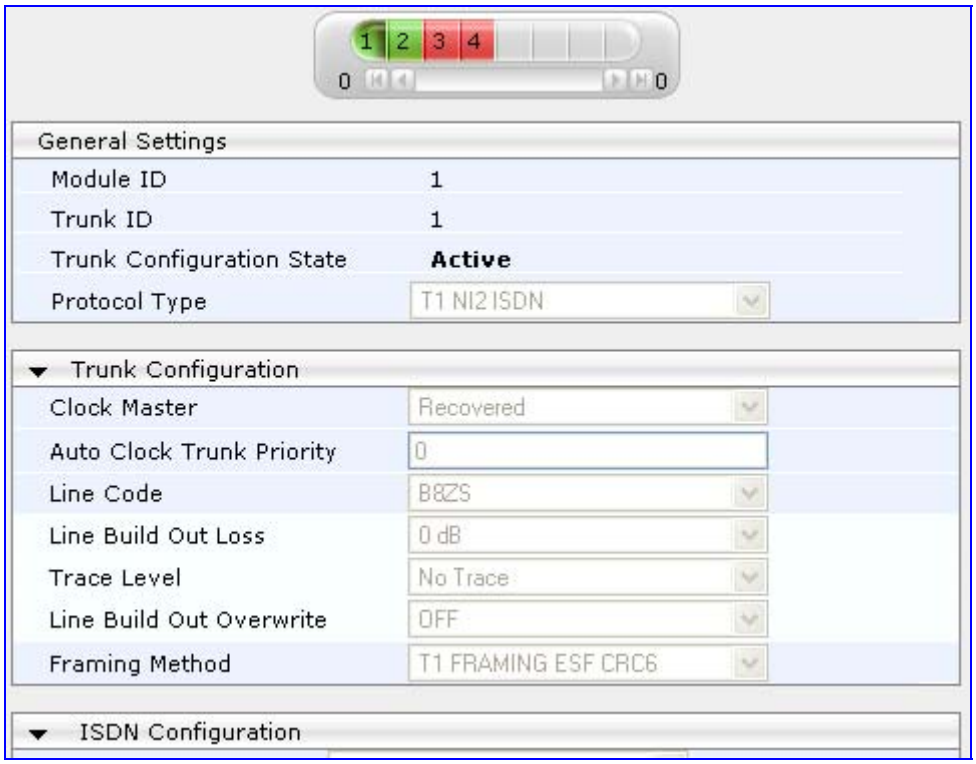
Notes:

- During trunk deactivation, trunk configuration cannot be performed.
- A stopped trunk cannot also be activated.

➢ **To configure the trunks:**

1. Open the 'Trunk Settings' page (**Configuration** tab > **PSTN Settings** menu > **Trunk Settings** page item).

Figure 3-54: Trunk Settings Page

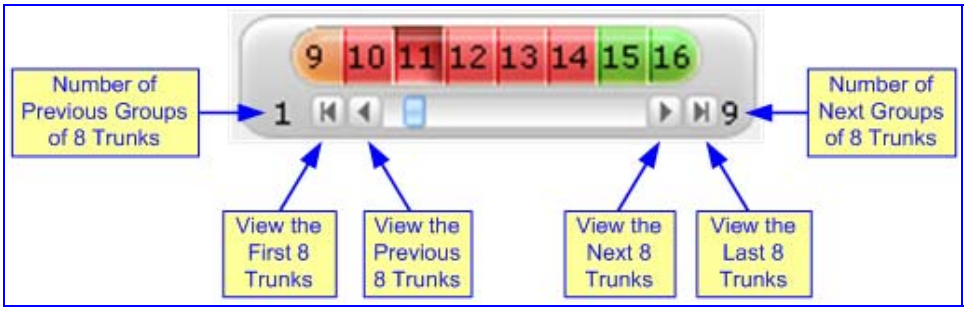


On the top of the page, a bar with Trunk number icons displays the status of each trunk, according to the following color codes:

- **Grey:** Disabled
- **Green:** Active
- **Yellow:** RAI alarm (also appears when you deactivate a Trunk by clicking the **Deactivate** button)
- **Red:** LOS/LOF alarm
- **Blue:** AIS alarm
- **Orange:** D-channel alarm (ISDN only)

2. Select the trunk that you want to configure, by clicking the desired Trunk number icon. The bar initially displays the first eight trunk number icons (i.e., trunks 1 through 8). To scroll through the trunk number icons (i.e., view the next/last or previous/first group of eight trunks), refer to the figure below:

Figure 3-55: Trunk Scroll Bar


9 10 11 12 13 14 15 16
1
9
Number of Previous Groups of 8 Trunks
Number of Next Groups of 8 Trunks
View the First 8 Trunks
View the Previous 8 Trunks
View the Next 8 Trunks
View the Last 8 Trunks

Note: If the Trunk scroll bar displays all the available trunks, the scroll bar buttons are unavailable.

After you have selected a trunk, the following is displayed:

- The read-only 'Module ID' field displays the module number to which the trunk belongs.
- The read-only 'Trunk ID' field displays the selected trunk number.
- The read-only 'Trunk Configuration State' displays the state of the trunk (e.g., 'Active' or 'Inactive').
- The parameters displayed in the page pertain to the selected trunk only.

3. Click the **Stop Trunk** button (located at the bottom of the page) to take the trunk out of service so that you can configure the currently grayed out (unavailable) parameters.(Skip this step if you want to configure parameters that are available when the trunk is active). The stopped trunk is indicated by the following:

- The 'Trunk Configuration State' field displays 'Inactive'.
- The **Stop Trunk** button is replaced by the **Apply Trunk Settings** button.

  (When all trunks are stopped, the **Apply to All Trunks** button also appears.)
- All the parameters are available and can be modified.

4. Configure the desired trunk parameters.
5. Click the **Apply Trunk Settings** button to apply the changes to the selected trunk (or click **Apply to All Trunks** to apply the changes to all trunks); the **Stop Trunk** button replaces **Apply Trunk Settings** and the 'Trunk Configuration State' displays 'Active'.
6. To save the changes to flash memory, refer to "Saving Configuration" on page 199.
7. To reset the device, refer to "Resetting the Device" on page 197.

Notes:

- If the 'Protocol Type' field displays 'NONE' (i.e., no protocol type selected) and no other trunks have been configured, after selecting a PRI protocol type, you must reset the device.
- The displayed parameters on the page depend on the protocol selected in the 'Protocol Type' field.
- All PRI trunks of the device must be of the same line type (i.e., E1 or T1). However, different variants of the same line type can be configured on different trunks, for example, E1 Euro ISDN and E1 CAS (subject to the constraints in the device's *Release Notes*).
- BRI trunks can operate with E1 or T1 trunks.
- If the trunk protocol type is CAS, you can assign or modify a dial plan (in the 'Dial Plan' field) and perform this without stopping the trunk.
- If the trunk can't be stopped because it provides the device's clock (assuming the device is synchronized with the E1/T1 clock), assign a different E1/T1 trunk to provide the device's clock or enable 'TDM Bus PSTN Auto Clock' in the 'TDM Bus Settings' page (refer to "Configuring TDM Bus Settings" on page 184).
- To delete a previously configured trunk, set the parameter 'Protocol Type' to 'None'.
- A trunk cannot be deactivated if it has been stopped (by clicking the **Stop** button).

3.3.4 Security Settings

The **Security Settings** menu allows you to configure various security settings. This menu contains the following page items:

- Web User Accounts (refer to "Configuring the Web User Accounts" on page 83)
- WEB & Telnet Access List (refer to "Configuring the Web and Telnet Access List" on page 85)
- Firewall Settings (refer to "Configuring the Firewall Settings" on page 86)
- Certificates (refer to "Configuring the Certificates" on page 89)
- General Security Settings (refer to "Configuring the General Security Settings" on page 93)
- IPSec Proposal Table (refer to "Configuring the IP Security Associations Table" on page 96)
- IPSec Association Table (refer to "Configuring the IP Security Proposal Table" on page 95)

3.3.4.1 Configuring the Web User Accounts

To prevent unauthorized access to the Web interface, two Web user accounts are available (primary and secondary) with assigned user name, password, and access level. When you login to the Web interface, you are requested to provide the user name and password of one of these Web user accounts. If the Web session is idle (i.e., no actions are performed) for more than five minutes, the Web session expires and you are once again requested to login with your user name and password. Up to five Web users can simultaneously open (log in to) a session on the device's Web interface.

Each Web user account is composed of three attributes:

- **User name and password:** enables access (login) to the Web interface.
- **Access level:** determines the extent of the access (i.e., availability of pages and read / write privileges). The available access levels and their corresponding privileges are listed in the table below:

Table 3-9: Web User Accounts Access Levels and Privileges

Access Level	Numeric Representation*	Privileges
Security Administrator	200	Read / write privileges for all pages.
Administrator	100	read / write privileges for all pages except security-related pages, which are read-only.
User Monitor	50	No access to security-related and file-loading pages; read-only access to the other pages. This read-only access level is typically applied to the secondary Web user account.
No Access	0	No access to any page.
* The numeric representation of the access level is used only to define accounts in a RADIUS server (the access level ranges from 1 to 255).		

The default attributes for the two Web user accounts are shown in the following table:

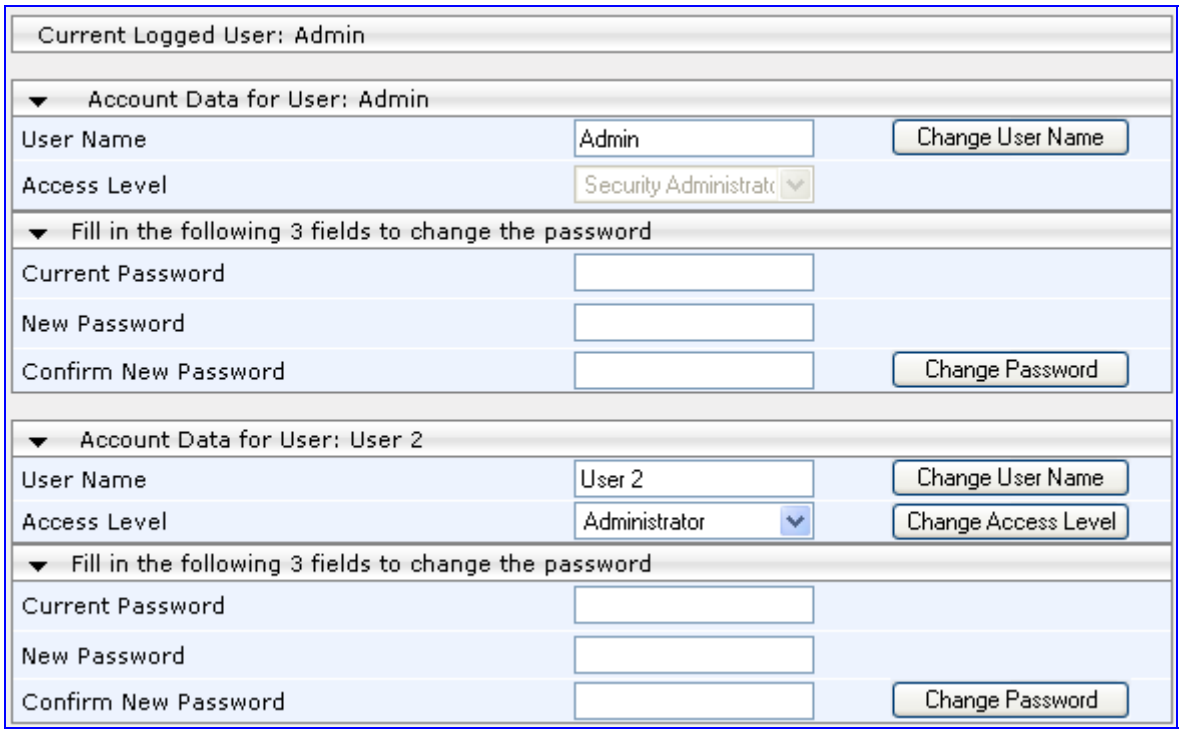
Table 3-10: Default Attributes for the Web User Accounts

Account / Attribute	User Name (Case-Sensitive)	Password (Case-Sensitive)	Access Level
Primary Account	Admin	Admin	Security Administrator **Note:** The Access Level cannot be changed for this account type.
Secondary Account	User	User	User Monitor

➢ **To change the Web user accounts attributes:**

1. Open the 'Web User Accounts' page (**Configuration** tab > **Security Settings** menu > **Web User Accounts** page item).

Figure 3-56: WEB User Accounts Page (for Users with 'Security Administrator' Privileges)



Note: If you are logged into the Web interface as the Security Administrator, both Web user accounts are displayed on the 'Web User Accounts' page (as shown above). If you are logged in with the secondary user account, only the details of the secondary account are displayed on the page.

2. To change the access level of the secondary account:
   a. From the 'Access Level' drop-down list, select the new access level.
   b. Click **Change Access Level**; the new access level is applied immediately.

Notes:

- The access level of the primary Web user account is 'Security Administrator', which cannot be modified.
- The access level of the secondary account can only be modified by the primary account user or a secondary account user with 'Security Administrator' access level.

3. To change the user name of an account, perform the following:
   a. In the field 'User Name', enter the new user name (maximum of 19 case-sensitive characters).
   b. Click **Change User Name**; if you are currently logged into the Web interface with this account, the 'Enter Network Password' dialog box appears, requesting you to enter the new user name.
4. To change the password of an account, perform the following:
   a. In the field 'Current Password', enter the current password.

b. In the fields 'New Password' and 'Confirm New Password', enter the new password (maximum of 19 case-sensitive characters).

c. Click **Change Password**; if you are currently logged into the Web interface with this account, the 'Enter Network Password' dialog box appears, requesting you to enter the new password.

Notes:

- For security, it's recommended that you change the default user name and password.
- A Web user with access level 'Security Administrator' can change all attributes of all the Web user accounts. Web users with an access level other than 'Security Administrator' can only change their own password and user name.
- To reset the two Web user accounts' user names and passwords to default, set the *ini* file parameter ResetWebPassword to 1.
- To access the Web interface with a different account, click the **Log off** button located on the toolbar, click any button or page item, and then re-access the Web interface with a different user name and password.
- You can set the entire Web interface to read-only (regardless of Web user account's access level), by using the *ini* file parameter DisableWebConfig (refer to "Web and Telnet Parameters" on page 260).
- Access to the Web interface can be disabled, by setting the *ini* file parameter DisableWebTask to 1. By default, access is enabled.
- You can define additional Web user accounts using a RADIUS server (refer to the *Product Reference Manual*).
- For secured HTTP connection (HTTPS), refer to the *Product Reference Manual*.

3.3.4.2 Configuring the Web and Telnet Access List

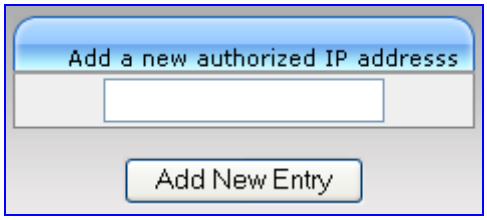
The 'Web & Telnet Access List' page is used to define up to ten IP addresses that are permitted to access the device's Web and Telnet interfaces. Access from an undefined IP address is denied. If no IP addresses are defined, this security feature is inactive and the device can be accessed from any IP address.

The Web and Telnet Access List can also be defined using the *ini* file parameter WebAccessList_x (refer to "Web and Telnet Parameters" on page 260).

➢ **To add authorized IP addresses for Web and Telnet interfaces access:**

1. Open the 'Web & Telnet Access List' page (**Configuration** tab > **Security Settings** menu > **Web & Telnet Access List** page item).

Figure 3-57: Web & Telnet Access List Page - Add New Entry



2. To add an authorized IP address, in the 'Add a New Authorized IP Address' field, enter the required IP address, and then click **Add New Address**; the IP address you entered is added as a new entry to the 'Web & Telnet Access List' table.

Figure 3-58: Web & Telnet Access List Table

3. To delete authorized IP addresses, select the Delete Row check boxes corresponding to the IP addresses that you want to delete, and then click **Delete Selected Addresses**; the IP addresses are removed from the table and these IP addresses can no longer access the Web and Telnet interfaces.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

Notes:

- The first authorized IP address in the list must be your PC's (terminal) IP address; otherwise, access from your PC is denied.
- Only delete your PC's IP address last from the 'Web & Telnet Access List' page. If it's deleted before the last, access from your PC is denied after it's deleted.

3.3.4.3 Configuring the Firewall Settings

The device provides an internal firewall, allowing you (the security administrator) to define network traffic filtering rules. You can add up to 50 ordered firewall rules.

The access list provides the following features:

- Block traffic from known malicious sources
- Only allow traffic from known friendly sources, and block all others
- Mix allowed and blocked network sources
- Limit traffic to a pre-defined rate (blocking the excess)
- Limit traffic to specific protocols, and specific port ranges on the device

For each packet received on the network interface, the table is scanned from the top down until a matching rule is found. This rule can either deny (*block*) or permit (*allow*) the packet. Once a rule in the table is located, subsequent rules further down the table are ignored. If the end of the table is reached without a match, the packet is accepted. For detailed information on the internal firewall, refer to the *Product Reference Manual*.

Note: You can also configure the firewall settings using the *ini* file table parameter AccessList (refer to "Security Parameters" on page 271).

➢ **To add firewall rules:**

1. Open the 'Firewall Settings' page (**Configuration** tab > **Security Settings** menu > **Firewall Settings** page item).

Figure 3-59: Firewall Settings Page

Edit Rule	Is Rule Active?	Source IP	Prefix Length	Local Port Range	Protocol	Packet Size	Byte rate	Burst Bytes	Action Upon Match	Match Count
1	Yes	mgmt.customer.com	32	0 - 80	tcp	0	0	0	Allow	0
2	Yes	192.0.0.0	8	0-65535	Any	0	40000	50000	ALLOW	0
3	Yes	10.31.4.0	24	4000-9000	Any	0	0	0	BLOCK	0
4	Yes	10.4.0.0	16	4000-9000	Any	0	0	0	BLOCK	0

2. In the 'Add' field, enter the index of the access rule that you want to add, and then click **Add**; a new firewall rule index appears in the table.
3. Configure the firewall rule's parameters according to the table below.
4. Click one of the following buttons:
   - **Apply:** saves the new rule (without activating it).
   - **Duplicate Rule:** adds a new rule by copying a selected rule.
   - **Activate:** saves the new rule and activates it.
   - **Delete:** deletes the selected rule.
5. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

The figure above shows the following access list settings:

- **Rule #1:** traffic from the host 'mgmt.customer.com' destined to TCP ports 0 to 80, is always allowed.
- **Rule #2:** traffic from the 192.xxx.yyy.zzz subnet, is limited to a rate of 40 Kbytes per second (with an allowed burst of 50 Kbytes). Note that the rate is specified in bytes, not bits, per second; a rate of 40000 bytes per second, nominally corresponds to 320 kbps.
- **Rule #3:** traffic from the subnet 10.31.4.xxx destined to ports 4000-9000 is always blocked, regardless of protocol.
- **Rule #4:** traffic from the subnet 10.4.xxx.yyy destined to ports 4000-9000 is always blocked, regardless of protocol.
- All other traffic is allowed

➢ **To edit a rule:**

1. In the 'Edit Rule' column, select the rule that you want to edit.
2. Modify the fields as desired.
3. Click the **Apply** button to save the changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

➢ **To activate a de-activated rule:**

1. In the 'Edit Rule' column, select the de-activated rule that you want to activate.
2. Click the **Activate** button; the rule is activated.

➢ **To de-activate an activated rule:**

1. In the 'Edit Rule' column, select the activated rule that you want to de-activate.
2. Click the **DeActivate** button; the rule is de-activated.

➢ **To delete a rule:**

1. Select the radio button of the entry you want to activate.
2. Click the **Delete Rule** button; the rule is deleted.
3. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

Table 3-11: Internal Firewall Parameters

Parameter	Description
Is Rule Active	A read-only field indicating whether the rule is active or not. **Note:** After device reset, all rules are active.
Source IP **[AccessList_Source_IP]**	IP address (or DNS name) of source network, or a specific host.
Prefix Length **[AccessList_PrefixLen]**	IP network mask. 32 for a single host, or the appropriate value for the source IP addresses. ▪ A value of 8 corresponds to IPv4 subnet class A (network mask of 255.0.0.0). ▪ A value of 16 corresponds to IPv4 subnet class B (network mask of 255.255.0.0). ▪ A value of 24 corresponds to IPv4 subnet class C (network mask of 255.255.255.0). The IP address of the sender of the incoming packet is trimmed in accordance with the prefix length (in bits) and then compared to the parameter 'Source IP'.
Local Port Range **[AccessList_Start_Port]** **[AccessList_End_Port]**	The destination UDP/TCP ports (on this device) to which packets are sent. The valid range is 0 to 65535. **Note:** When the protocol type isn't TCP or UDP, the entire range must be provided.

Parameter	Description
Protocol **[AccessList_Protocol]**	The protocol type (e.g., UDP, TCP, ICMP, ESP or 'Any'), or the IANA protocol number (in the range of 0 (Any) to 255). **Note:** This field also accepts the abbreviated strings 'SIP' and 'HTTP'. Specifying these strings implies selection of the TCP or UDP protocols, and the appropriate port numbers as defined on the device.
Packet Size **[AccessList_Packet_Size]**	Maximum allowed packet size. The valid range is 0 to 65535. **Note:** When filtering fragmented IP packets, this field relates to the overall (re-assembled) packet size, and not to the size of each fragment.
Byte Rate **[AccessList_Byte_Rate]**	Expected traffic rate (bytes per second).
Burst Bytes **[AccessList_Byte_Burst]**	Tolerance of traffic rate limit (number of bytes).
Action Upon Match **[AccessList_Allow_Type]**	Action upon match (i.e., 'Allow' or 'Block').
Match Count **[AccessList_MatchCount]**	A read-only field providing the number of packets accepted / rejected by the specific rule.

3.3.4.4 Configuring the Certificates

The 'Certificates' page is used for both HTTPS and SIP TLS secure communication:

- Replacing the server certificate (refer to "Server Certificate Replacement" on page 89)
- Replacing the client certificates (refer to "Client Certificates" on page 92)
- Regenerating Self-Signed Certificates (refer to "Self-Signed Certificates" on page 92)
- Updating the private key (using HTTPSPkeyFileName, as described in the *Product Reference Manual*).

3.3.4.4.1 Server Certificate Replacement

The device is supplied with a working Secure Socket Layer (SSL) configuration consisting of a unique self-signed server certificate. If an organizational Public Key Infrastructure (PKI) is used, you may wish to replace this certificate with one provided by your security administrator.

➢ **To replace the device's self-signed certificate:**

1. Your network administrator should allocate a unique DNS name for the device (e.g., dns_name.corp.customer.com). This DNS name is used to access the device and should therefore, be listed in the server certificate.
2. If the device is operating in HTTPS mode, then set the parameter 'Secured Web Connection (HTTPS)' to 'HTTP and HTTPS' (0) (refer to "Configuring the General Security Settings" on page 93) to ensure you have a method of accessing the device in case the new certificate doesn't work. Restore the previous setting after testing the configuration.

3. Open the 'Certificates Signing Request' page (**Configuration** tab > **Security Settings** menu > **Certificates** page item).

Figure 3-60: Certificates Signing Request Page

Certificate Signing Request
Subject Name
Generate CSR
Copy the certificate signing request and send it to your Certification Authority for signing.
Press the button "Generate self-signed" to create a self-signed certificate using the subject name provided above.
Important: this is a lengthy operation, during this time the device will be out of service.
After the operation is complete, save configuration and reset the device.
Generate self-signed
Certificate Files
Send "Server Certificate" file from your computer to the device
Browse...
Send File
Send "Trusted Root Certificate Store" file from your computer to the device
Browse...
Send file
Send "Private Key" file from your computer to the device
Browse...
Send file
Note: Replacing the private key is not recommended but if it's done, it should be over a physically-secure network link.

4. In the 'Subject Name' field, enter the DNS name, and then click **Generate CSR**. A textual certificate signing request that contains the SSL device identifier is displayed.

5. Copy this text and send it to your security provider. The security provider (also known as Certification Authority or CA) signs this request and then sends you a server certificate for the device.

6. Save the certificate to a file (e.g., cert.txt). Ensure that the file is a plain-text file containing the 'BEGIN CERTIFICATE' header, as shown in the example of a Base64-Encoded X.509 Certificate below:

```
-----BEGIN CERTIFICATE-----
MIIDkzCCAnugAwIBAgIEAgAAADANBgkqhkiG9w0BAQQFADA/MQswCQYDVQQGEwJGUj
ETMBEGA1UEChMKQ2VydGlwb3N0ZTEbMBkGA1UEAxMSQ2VydGlwb3N0ZSBTZXJ2ZXVy
MB4XDTk4MDYyNDA4MDAwMFoXDTE4MDYyNDA4MDAwMFowPzELMAkGA1UEBhMCRlIxEz
ARBgNVBAoTCkNlcnRpcG9zdGUxGzAZBgNVBAMTEkNlcnRpcG9zdGUgU2VydmV1cjCC
ASEwDQYJKoZIhvcNAQEBBQADggEOADCCAQkCggEAPqd4MziR4spWldGRx8bQrhZkon
WnNm`+Yhb7+4Q67ecf1janH7GcN/SXsfx7jJpreWULf7v7Cvpr4R7qIJcmdHIntmf7
JPM5n6cDBv17uSW63er7NkVnMFHwK1QaGFLMybFkzaeGrvFm4k3lRefiXDmuOe+FhJ
gHYezYHf44LvPRPwhSrzi9+Aq3o8pWDguJuZDIUP1F1jMa+LPwvREXfFcUW+w==
-----END CERTIFICATE-----
```

7. In the 'Certificates Files' group, click the **Browse** button corresponding to 'Send Server Certificate...', navigate to the cert.txt file, and then click **Send File**.

8. When the loading of the certificate is complete, save the configuration (refer to "Saving Configuration" on page 199) and restart the device; the Web interface uses the provided certificate.

9. If the device was originally operating in HTTPS mode and you disabled it in Step 2, then return it to HTTPS by setting the parameter 'Secured Web Connection (HTTPS)' to 'HTTPS Only' (1) (refer to "Configuring the General Security Settings" on page 93).

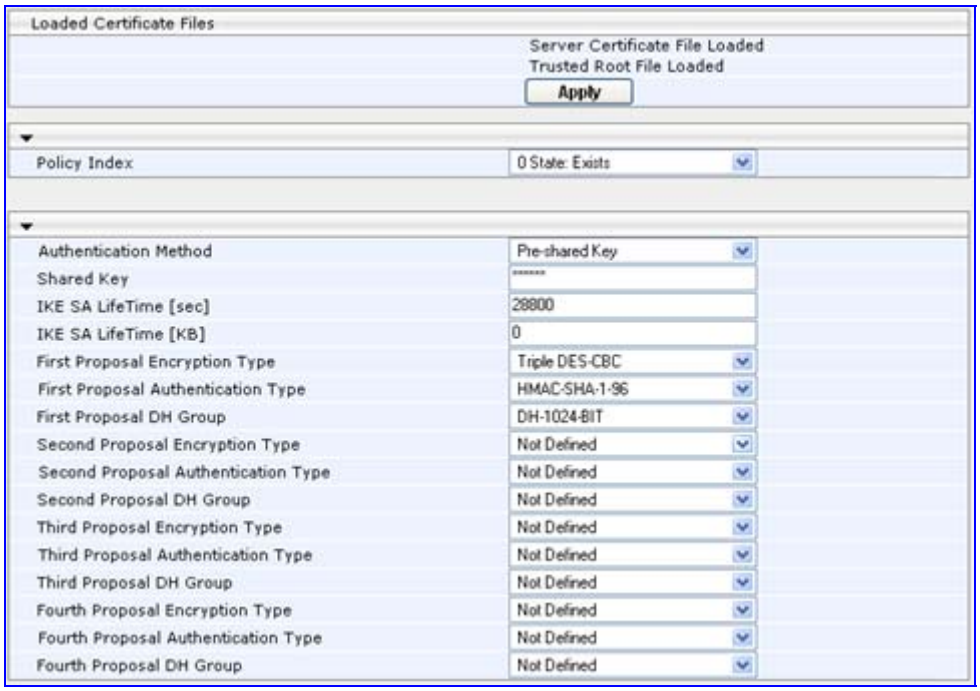
Notes:

- The certificate replacement process can be repeated when necessary (e.g., the new certificate expires).
- It is possible to use the IP address of the device (e.g., 10.3.3.1) instead of a qualified DNS name in the Subject Name. This is not recommended since the IP address is subject to changes and may not uniquely identify the device.
- The server certificate can also be loaded via *ini* file using the parameter HTTPSCertFileName.

➢ **To apply the loaded certificate for IPSec negotiations:**

1. Open the 'IKE Table' page (refer to "Configuring the IP Security Associations Table" on page 95); the 'Loaded Certificates Files' group lists the newly uploaded certificates, as shown below:

Figure 3-61: IKE Table Listing Loaded Certificate Files



2. Click the **Apply** button to load the certificates; future IKE negotiations are now performed using the new certificates.

3.3.4.4.2 Client Certificates

By default, Web servers using SSL provide one-way authentication. The client is certain that the information provided by the Web server is authentic. When an organizational PKI is used, two-way authentication may be desired: both client and server should be authenticated using X.509 certificates. This is achieved by installing a client certificate on the managing PC, and loading the same certificate (in base64-encoded X.509 format) to the device's Trusted Root Certificate Store. The Trusted Root Certificate file should contain both the certificate of the authorized user and the certificate of the CA.

Since X.509 certificates have an expiration date and time, the device must be configured to use NTP (refer to "Simple Network Time Protocol Support" on page 607) to obtain the current date and time. Without the correct date and time, client certificates cannot work.

➢ **To enable two-way client certificates:**

1. Set the parameter 'Secured Web Connection (HTTPS)' to 'HTTPS Only' (0) in "Configuring the General Security Settings" on page 93 to ensure you have a method of accessing the device in case the client certificate doesn't work. Restore the previous setting after testing the configuration.
2. Open the 'Certificates Signing Request' page (refer to "Server Certificate Replacement" on page 89).
3. In the 'Certificates Files' group, click the **Browse** button corresponding to 'Send "Trusted Root Certificate Store" file ...', navigate to the file, and then click **Send File**.
4. When the operation is complete, set the *ini* file parameter HTTPSRequireClientCertificates to 1.
5. Save the configuration (refer to "Saving Configuration" on page 199), and then restart the device.

When a user connects to the secured Web server:

- If the user has a client certificate from a CA that is listed in the Trusted Root Certificate file, the connection is accepted and the user is prompted for the system password.
- If both the CA certificate and the client certificate appear in the Trusted Root Certificate file, the user is not prompted for a password (thus, providing a single-sign-on experience - the authentication is performed using the X.509 digital signature).
- If the user doesn't have a client certificate from a listed CA, or doesn't have a client certificate at all, the connection is rejected.

Notes:

- The process of installing a client certificate on your PC is beyond the scope of this document. For more information, refer to your Web browser or operating system documentation, and/or consult your security administrator.
- The root certificate can also be loaded via *ini* file using the parameter HTTPSRootFileName.
- You can enable Online Certificate Status Protocol (OCSP) on the device to check whether a peer's certificate has been revoked by an OCSP server. For further information, refer to the *Product Reference Manual*.

3.3.4.4.3 Self-Signed Certificates

The device is shipped with an operational, self-signed server certificate. The subject name for this default certificate is 'ACL_nnnnnnn', where *nnnnnnn* denotes the serial number of the device. However, this subject name may not be appropriate for production and can be changed while still using self-signed certificates.

➢ **To change the subject name and regenerate the self-signed certificate:**

1. Before you begin, ensure the following:
   - You have a unique DNS name for the device (e.g., dns_name.corp.customer.com). This name is used to access the device and should therefore, be listed in the server certificate.
   - No traffic is running on the device. The certificate generation process is disruptive to traffic and should be executed during maintenance time.
2. Open the 'Certificates' page (refer to "Server Certificate Replacement" on page 89).
3. In the 'Subject Name' field, enter the fully-qualified DNS name (FQDN) as the certificate subject, and then click **Generate Self-signed**; after a few seconds, a message appears displaying the new subject name.
4. Save configuration (refer to "Saving Configuration" on page 199), and then restart the device for the new certificate to take effect.

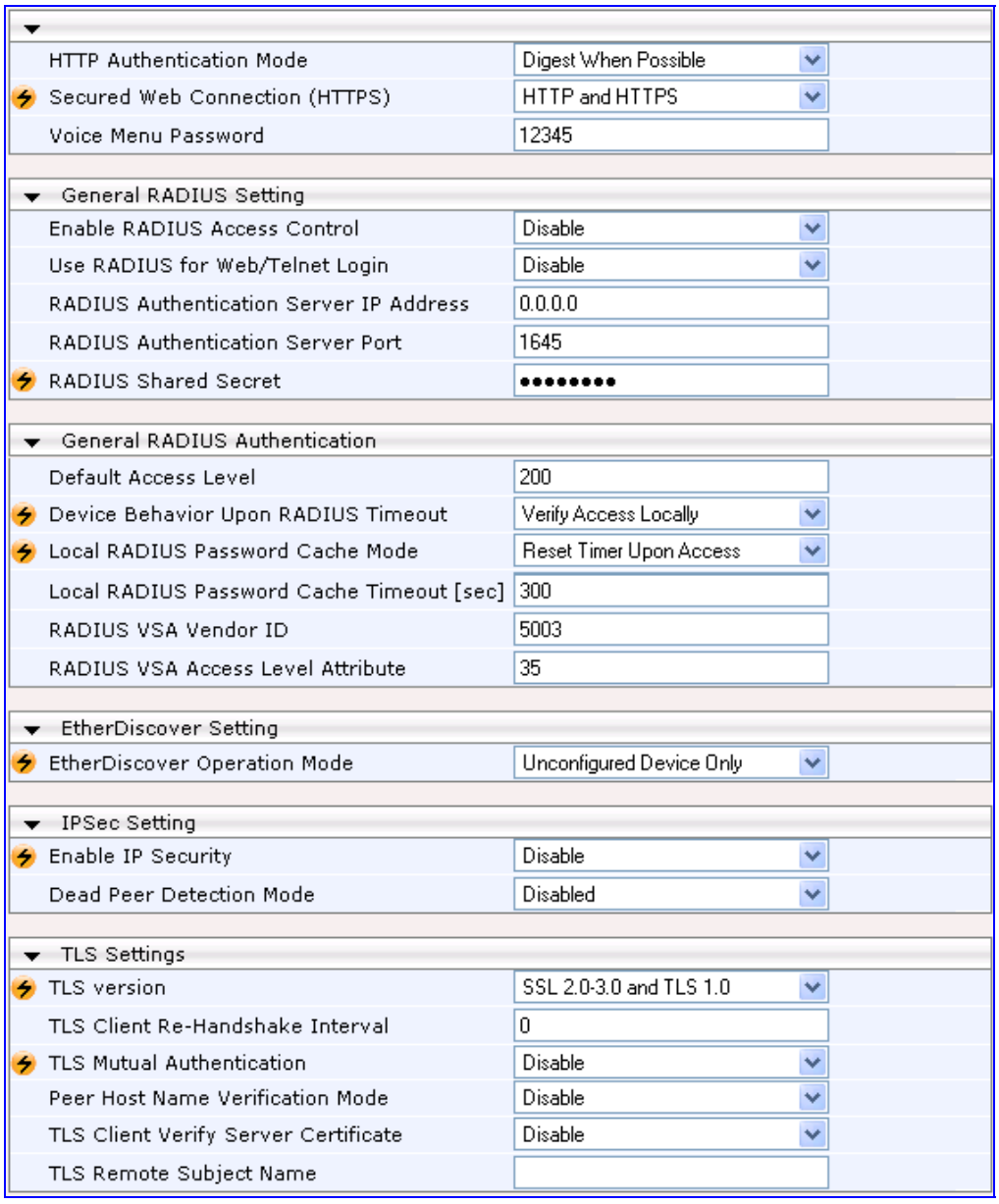
3.3.4.5 Configuring the General Security Settings

The 'General Security Settings' page is used to configure various security features. For a description of the parameters appearing on this page, refer "Configuration Parameters Reference" on page 247.

➢ **To configure the general security parameters:**

1. Open the 'General Security Settings' page (**Configuration** tab > **Security Settings** menu > **General Security Settings** page item).

Figure 3-62: General Security Settings Page



2. Configure the parameters as required.
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

3.3.4.6 Configuring the IP Security Proposal Table

The 'IP Security Proposals Table' page is used to configure Internet Key Exchange (IKE) with up to four proposal settings. Each proposal defines an encryption algorithm, an authentication algorithm, and a Diffie-Hellman group identifier. The same set of proposals apply to both Main mode and Quick mode.

Note: You can also configure the IP Security Proposals table using the *ini* file table parameter IPsecProposalTable (refer to "Security Parameters" on page 271).

➢ **To configure IP Security Proposals:**

1. Open the 'IP Security Proposals Table' page (**Configuration** tab > **Security Settings** menu > **IPSec Proposal Table**).

Figure 3-63: IP Security Proposals Table

Index	Encryption Algorithm	Authentication Algorithm	Diffie Hellman Group
0	AES	HMAC SHA1 96	Group 2 [1024 Bits]
1	3DES CBC	HMAC SHA1 96	Group 2 [1024 Bits]

In the figure above, two proposals are defined:

- Proposal 0: AES, SHA1, DH group 2
- Proposal 1: 3DES, SHA1, DH group 2

Note that with this configuration, neither DES nor MD5 can be negotiated

2. Select an Index, click **Edit**, and then modify the proposal as required.
3. Click **Apply**.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

To delete a proposal, select the relevant Index number, and then click **Delete**.

Table 3-12: IP Security Proposals Table Configuration Parameters

Parameter Name	Description
Encryption Algorithm **[IPsecProposalTable_EncryptionAlgorithm]**	Determines the encryption (privacy) algorithm. ▪ **[0]** NONE ▪ **[1]** DES CBC ▪ **[2]** 3DES CBC ▪ **[3]** AES (default)
Authentication Algorithm **[IPsecProposalTable_AuthenticationAlgorithm]**	Determines the message authentication (integrity) algorithm. ▪ **[0]** NONE ▪ **[2]** HMAC SHA1 96 ▪ **[4]** HMAC MD5 96 (default)

Parameter Name	Description
Diffie Hellman Group **[IPsecProposalTable_DHGroup]**	Determines the length of the key created by the DH protocol for up to four proposals. For the *ini* file parameter, *X* depicts the proposal number (0 to 3). ▪ **[0]** Group 1 (768 Bits) = DH-786-Bit ▪ **[1]** Group 2 (1024 Bits) (default) = DH-1024-Bit

If no proposals are defined, the default settings (shown in the following table) are applied.

Table 3-13: Default IPSec/IKE Proposals

Proposal	Encryption	Authentication	DH Group
Proposal 0	3DES	SHA1	Group 2 (1024 bit)
Proposal 1	3DES	MD5	Group 2 (1024 bit)
Proposal 2	3DES	SHA1	Group 1 (786 bit)
Proposal 3	3DES	MD5	Group 1 (786 bit)

3.3.4.7 Configuring the IP Security Associations Table

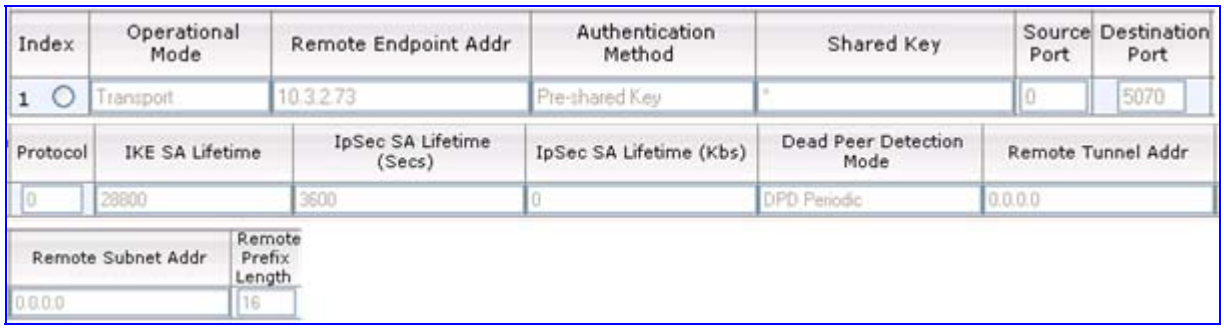
The 'IP Security Associations Table' page allows you to configure up to 20 peers (hosts or networks) for IP security (IPSec)/IKE. Each of the entries in the IPSec Security Association table controls both Main Mode and Quick Mode configuration for a single peer

Note: You can also configure the IP Security Associations table using the *ini* file table parameter IPsecSATable (refer to "Security Parameters" on page 271).

➢ **To configure the IPSec Association table:**

1. Open the 'IP Security Associations Table' page (**Configuration** tab > **Security Settings** menu > **IPSec Association Table**). (Due to the length of the table, the figure below shows sections of this table.)

Figure 3-64: IP Security Associations Table Page



2. Add an Index or select the Index rule you want to edit.
3. Configure the rule according to the table below.

4. Click **Apply**; the rule is applied on-the-fly.
5. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

Table 3-14: IP Security Associations Table Configuration Parameters

Parameter Name	Description
Operational Mode **[IPsecSATable_IPsecMode]**	Defines the IPSec mode of operation. ▪ **[0]** Transport (default) ▪ **[1]** Tunneling
Remote Endopint **[IPsecSATable_RemoteEndpointAddressOrName]**	Defines the IP address or DNS host name of the peer. **Note:** This parameter is applicable only if the Operational Mode is set to Transport.
Authentication Method **[IPsecSATable_AuthenticationMethod]**	Selects the method used for peer authentication during IKE main mode. ▪ **[0]** Pre-shared Key (default) ▪ **[1]** RSA Signature = in X.509 certificate **Note:** For RSA-based authentication, both peers must be provisioned with certificates signed by a common CA. For more information on certificates refer to "Server Certificate Replacement" on page 89.
Shared Key **[IPsecSATable_SharedKey]**	Defines the pre-shared key (in textual format). Both peers must use the same pre-shared key for the authentication process to succeed. **Notes:** ▪ This parameter is applicable only if the Authentication Method parameter is set to pre-shared key. ▪ The pre-shared key forms the basis of IPSec security and therefore, it should be handled with care (the same as sensitive passwords). It is not recommended to use the same pre-shared key for several connections. ▪ Since the *ini* file is plain text, loading it to the device over a secure network connection is recommended. Use a secure transport such as HTTPS, or a direct crossed-cable connection from a management PC. ▪ After it is configured, the value of the pre-shared key cannot be retrieved.
Source Port **[IPsecSATable_SourcePort]**	Defines the source port to which this configuration applies. The default value is 0 (i.e., any port).
Destination Port **[IPsecSATable_DestPort]**	Defines the destination port to which this configuration applies. The default value is 0 (i.e., any port).

Parameter Name	Description
Protocol **[IPsecSATable_Protocol]**	Defines the protocol type to which this configuration applies. Standard IP protocol numbers, as defined by the Internet Assigned Numbers Authority (IANA) should be used, for example: ▪ 0 = Any protocol (default) ▪ 17 = UDP ▪ 6 = TCP
IKE SA Lifetime **[IPsecSATable_Phase1SaLifetimeInSec]**	Determines the duration (in seconds) for which the negotiated IKE SA (Main mode) is valid. After this time expires, the SA is re-negotiated. **Note:** Main mode negotiation is a processor-intensive operation; for best performance, do not set this parameter to less than 28,800 (i.e., eight hours). The default value is 0 (i.e., unlimited).
IPSec SA Lifetime (sec) **[IPsecSATable_Phase2SaLifetimeInSec]**	Determines the duration (in seconds) for which the negotiated IPSec SA (Quick mode) is valid. After this time expires, the SA is re-negotiated. The default value is 0 (i.e., unlimited). **Note:** For best performance, a value of 3,600 (i.e., one hour) or more is recommended.
IPSec SA Lifetime (Kbs) **[IPsecSATable_Phase2SaLifetimeInKB]**	Determines the maximum volume of traffic (in kilobytes) for which the negotiated IPSec SA (Quick mode) is valid. After this specified volume is reached, the SA is re-negotiated. The default value is 0 (i.e., the value is ignored).
Dead Peer Detection Mode **[IPsecSATable_DPDmode]**	Configures dead peer detection (DPD), according to RFC 3706. ▪ **[0]** DPD Disabled (default) ▪ **[1]** DPD Periodic = DPD is enabled with message exchanges at regular intervals ▪ **[2]** DPD on demand = DPD is enabled with on-demand checks - message exchanges as needed (i.e., before sending data to the peer). If the liveliness of the peer is questionable, the device sends a DPD message to query the status of the peer. If the device has no traffic to send, it never sends a DPD message. **Note:** For detailed information on DPD, refer to the P*roduct Reference Manual*.
Remote Tunnel Addr **[IPsecSATable_RemoteTunnelAddress]**	Defines the IP address of the peer router. **Note:** This parameter is applicable only if the Operational Mode is set to Tunnel.
Remote Subnet Addr **[IPsecSATable_RemoteSubnetIPAddress]**	Defines the IP address of the remote subnetwork. Together with the Prefix Length parameter (below), this parameter defines the network with which the IPSec tunnel allows communication. **Note:** This parameter is applicable only if the Operational Mode is set to Tunnel.

Parameter Name	Description
Remote Prefix Length **[IPsecSATable_RemoteSubnetPrefixLength]**	Defines the prefix length of the Remote Subnet IP Address parameter (in bits). The prefix length defines the subnet class of the remote network. A prefix length of 16 corresponds to a Class B subnet (255.255.0.0); a prefix length of 24 corresponds to a Class C subnet (255.255.255.0). **Note:** This parameter is applicable only if the Operational Mode is set to Tunnel.

3.3.5 Protocol Configuration

The **Protocol Configuration** menu allows you to configure the device's SIP parameters and contains the following submenus:

- Media Realm Configuration (refer to “Configuring Media Realms” on page 99)
- Applications Enabling (refer to “Enabling Applications” on page 101)
- Trunk Group (refer to ‘Trunk Group’ on page 102)
- Protocol Definition (refer to "Protocol Definition" on page 107)
- Application Network Settings (refer to “Application Network Setting” on page 110)
- Proxies, Registration, IP Groups (refer to “Proxies, Registrations, IP Groups” on page 113)
- Coders And Profile Definitions (refer to "Coders and Profile Definitions" on page 127)
- SIP Advanced Parameters (refer to "SIP Advanced Parameters" on page 135)
- Manipulation Tables (refer to “Manipulation Tables” on page 142)
- Routing Tables (refer to "Routing Tables" on page 155)
- Endpoint Settings (refer to “Endpoint Settings” on page 170)
- Digital Gateway (refer to “Configuring Digital Gateway Parameters” on page 178)
- IP Media (refer to “Configuring the IPmedia Parameters” on page 179)
- SAS (refer to “SAS Parameters” on page 179)

3.3.5.1 Configuring Media Realms

The 'SIP Media Realm Table' page allows you to define a pool of up to 16 media interfaces, termed *Media Realms*. This table allows you to divide a Media-type interface (defined in the 'Multiple Interface' table - refer to "Configuring the Multiple Interface Table" on page 61) into several realms, where each realm is specified by a UDP port range. Once created, the Media Realm can be assigned to other elements such as an IP Group (in the 'IP Group' table).

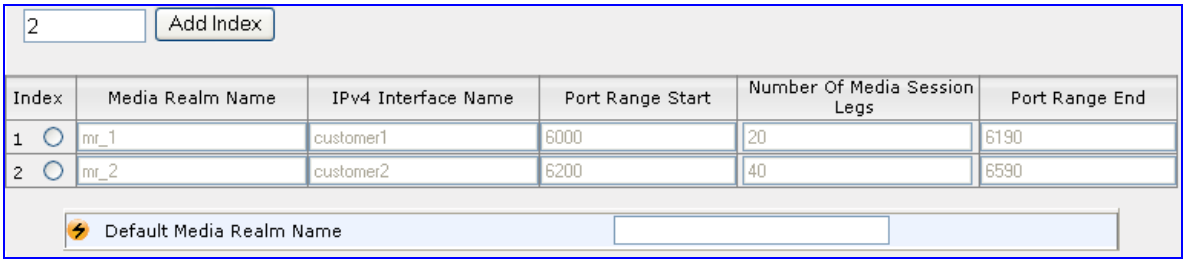
Notes:

- Up to 16 Media Realms can be configured in this table. However, only up to 8 Media Realms can be used by the device (as a maximum of 8 IP Groups can be configured).
- You can also configure the Media Realm table using the *ini* file table parameter CpMediaRealm.
- For this parameter to take effect, a device reset is required.

➢ **To define a Media Realm:**

1. Open the 'SIP Media Realm Table' page (**Configuration** tab > **Protocol Configuration** menu > **Media Realm Configuration** page item).

Figure 3-65: SIP Media Realm Table Page



2. In the 'Add Index' field, enter the required index number, and then click **Add Index**.
3. Configure the parameters according to the table below.
4. Click **Apply**; the entry is validated.
5. Click **Submit**.
6. Reset the device to save the changes to flash memory (refer to 'Saving Configuration' on page 199.

Table 3-15: SIP Media Realm Table Parameters

Parameter	Description
Media Realm Name **[CpMediaRealm_MediaRealmName]**	Defines an arbitrary, identifiable name for the Media Realm. The valid value is a string of up to 40 characters. **Notes:** ▪ This parameter is mandatory. ▪ The name assigned to the Media Realm must be unique. ▪ This Media Realm name is used in the 'IP Groups' table.

Parameter	Description
IPv4 Interface Name **[CpMediaRealm_IPv4IF]**	Associates the IPv4 interface to the Media Realm. **Note:** The name of this interface must be exactly as configured in the 'Multiple Interface' table (InterfaceTable parameter).
Port Range Start **[CpMediaRealm_PortRangeStart]**	Defines the starting port for the range of Media interface UDP ports. **Notes:** ▪ You must either configure all media realms with port ranges or without (not some with and some without). ▪ The available UDP port range is calculated as follows: BaseUDPPort (parameter) to BaseUDPPort plus 3290. For example, if BaseUDPPort is 6000 (default), then the available port range is 6000-9290. ▪ Port ranges over 60000 must not be used. ▪ Ranges of Media Realm ports must not overlap.
Number of Media Session Legs **[CpMediaRealm_MediaSessionLeg]**	Defines the number of media sessions associated with the range of ports. This is the number of media sessions available in the port range. For example, 100 ports correspond to 10 media sessions, since ports are allocated in chunks of 10.
Port Range End **[CpMediaRealm_PortRangeEnd]**	Read-only field displaying the ending port for the range of Media interface UDP ports. This field is calculated by adding the 'Media Session Leg' field (multiplied by the port chunk size) to the 'Port Range Start' field. A value appears once a row has been successfully added to the table.
Default Media Realm Name **[cpDefaultMediaRealmName]**	Defines any one of the Media Realms listed in this table as the default Media Realm. This default Media Realm is used when no Media Realm is configured for an IP Group for a specific call. The valid range is a string of up to 39 characters. **Notes:** ▪ If this parameter is not configured, then the first Media Realm configured in the SIP Media Realm table (cpMediaRealm) is used as the default Media Realm. ▪ If the SIP Media Realm table is not configured, then the default Media Realm includes all the device's media interfaces.

3.3.5.2 Enabling Applications

The 'Applications Enabling' page allows you to enable the following applications:

- Stand-Alone Survivability (SAS) application
- IP-to-IP (IP2IP) application

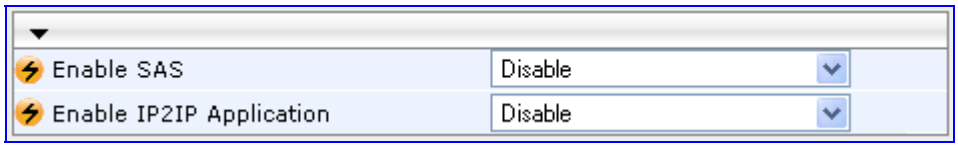
Notes:

- This page displays the application only if the device is installed with the relevant Software Upgrade Key supporting the application (refer to "Loading a Software Upgrade Key" on page 202).
- The IP2P application is applicable only to Mediant 1000.
- For enabling an application, a device reset is required.

➢ **To enable an application:**

1. Open the 'Applications Enabling' page (**Configuration** tab > **Protocol Configuration** menu > **Applications Enabling** page item).

Figure 3-66: Applications Enabling Page



2. Save the changes to the device's flash memory and then reset the device (refer to "Saving Configuration" on page 199).

3.3.5.3 Trunk Group

The **Trunk Group** submenu allows you to configure groups of channels called Trunk Groups. This submenu includes the following page items:

- Trunk Group Table (refer to “Configuring the Trunk Group Table” on page 102)
- Trunk Group Settings (refer to “Configuring Trunk Group Settings" on page 104)

3.3.5.3.1 Configuring the Trunk Group Table

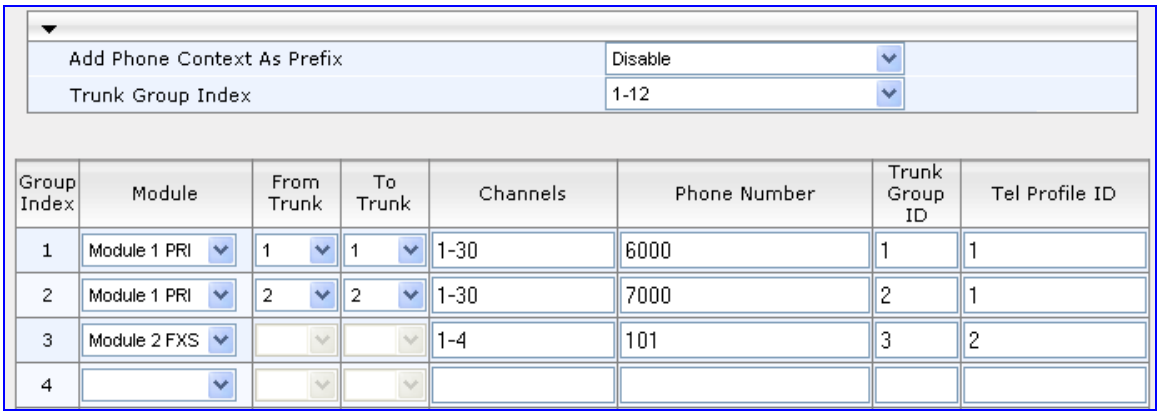
The 'Trunk Group Table' page allows you to define up to 120 Trunk Groups. This table enables the device's channels by assigning them telephone numbers and other optional attributes (e.g., Trunk Group IDs and Tel Profiles). Channels that are not defined in this table are disabled. Trunk Groups are used for routing calls (Tel-to-IP and IP-to-Tel) for the channels associated with the Trunk Group.

Note: You can also configure Trunk Groups using the *ini* file table parameter TrunkGroup_x to (refer to "Number Manipulation and Routing Parameters" on page 428).

➢ **To configure the Trunk Group Table:**

1. Open the 'Trunk Group Table' page (**Configuration** tab > **Protocol Configuration** menu > **Trunk Group** submenu > **Trunk Group** page item).

Figure 3-67: Trunk Group Table Page



2. Configure the Trunk Group according to the table below**.**
3. Click the **Submit** button to save your changes.
4. To save the changes to the flash memory, refer to "Saving Configuration" on page 199.

Table 3-16: Trunk Group Table Parameters

Parameter	Description
Module **[TrunkGroup_**Module]	The module type for which you want to define the Trunk Group.
From Trunk **[TrunkGroup_FirstTrunkId]**	Starting physical Trunk number in the Trunk Group. The number of listed Trunks depends on the device's hardware configuration. **Note:** This parameter is applicable only to PRI and BRI modules.
To Trunk **[TrunkGroup_LastTrunkId]**	Ending physical Trunk number in the Trunk Group. The number of listed Trunks depends on the device's hardware configuration. **Note:** This parameter is applicable only to PRI and BRI modules.
Channels **[TrunkGroup_FirstBChannel]**, **[TrunkGroup_LastBChannel]**	The device's channels/ports (analog module) or Trunk B-channels (digital module). To enable channels, enter the channel numbers. You can enter a range of channels by using the format **[n-**m], where *n* represents the lower channel number and *m* the higher channel number. For example, **[1-**4] specifies channels 1 through 4. **Notes:** ▪ The number of defined channels must not exceed the maximum number of the Trunk's B-channels. ▪ To represent all the Trunk's B-channels, enter a single asterisk (*).

Parameter	Description
Phone Number **[TrunkGroup_FirstPhoneNumber]**	The telephone number that is assigned to the channel. For a range of channels, enter only the first telephone number. Subsequent channels are assigned the next consecutive telephone number. For example, if you enter 400 for channels 1 to 4, then channel 1 is assigned phone number 400, channel 2 is assigned phone number 401, and so on. These numbers are also used for channel allocation for IP-to-Tel calls if the Trunk Group's 'Channel Select Mode' is set to 'By Dest Phone Number'. **Notes:** ▪ If this field includes alphabetical characters and the phone number is defined for a range of channels (e.g., 1-4), then the phone number must end with a number (e.g., 'user1'). ▪ This field is optional for BRI/PRI interfaces. The logical numbers defined in this field are used when an incoming PSTN/PBX call doesn't contain the calling number or called number (the latter being determined by the parameter ReplaceEmptyDstWithPortNumber). These numbers are used to replace them.
Trunk Group ID **[TrunkGroup_TrunkGroupNum]**	The Trunk Group ID (0-119) assigned to the corresponding channels. The same Trunk Group ID can be assigned to more than one group of channels. The Trunk Group ID is used to define a group of common channel behavior that are used for routing IP-to-Tel calls. If an IP-to-Tel call is assigned to a Trunk Group, the IP call is routed to the channel(s) pertaining to that Trunk Group ID. **Notes:** ▪ Once you have defined a Trunk Group, you must configure the parameter PSTNPrefix (Inbound IP Routing Table) to assign incoming IP calls to the appropriate Trunk Group. If you do not configure this, calls cannot be established. ▪ You can define the method for which calls are assigned to channels within Trunk Groups, using the parameter TrunkGroupSettings.
Tel Profile ID **[TrunkGroup_ProfileId]**	The Tel Profile ID assigned to the channels pertaining to the Trunk Group. **Note:** For configuring Tel Profiles, refer to the parameter TelProfile.

3.3.5.3.2 Configuring Trunk Group Settings

The 'Trunk Group Settings' page allows you to configure the settings of up to 120 Trunk Groups. These Trunk Groups are configured in the 'Trunk Group Table' page (refer to "Configuring the Trunk Group Table" on page 102). This page allows you to select the method for which IP-to-Tel calls are assigned to channels within each Trunk Group. If no method is selected (for a specific Trunk Group), the setting of the global parameter, ChannelSelectMode takes effect. In addition, this page defines the method for registering Trunk Groups to selected Serving IP Group IDs (if defined).

Note: You can also configure the 'Trunk Group Settings' table using the *ini* file table parameter TrunkGroupSettings (refer to "Number Manipulation and Routing Parameters" on page 428).

➢ **To configure the Trunk Group Settings table:**

1. Open the 'Trunk Group Settings' page (**Configuration** tab > **Protocol Configuration** menu > **Trunk Group** submenu > **Trunk Group Settings** page item).

Figure 3-68: Trunk Group Settings Page

2. From the 'Routing Index' drop-down list, select the range of entries that you want to edit.
3. Configure the Trunk Group according to the table below.
4. Click the **Submit** button to save your changes.
5. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

An example is shown below of a REGISTER message for registering endpoint "101" using registration Per Endpoint mode. The "SipGroupName" in the request URI is taken from the IP Group table.

```
REGISTER sip:SipGroupName SIP/2.0
Via: SIP/2.0/UDP 10.33.37.78;branch=z9hG4bKac862428454
From: <sip:101@GatewayName>;tag=1c862422082
To: <sip:101@GatewayName>
Call-ID: 990797706251200023282 5@10.33.37.78
CSeq: 3 REGISTER
Contact: <sip:101@10.33.37.78>;expires=3600
Expires: 3600
User-Agent: Sip-Gateway/v.6.00A.008.002
Content-Length: 0
```

Table 3-17: Trunk Group Settings Parameters

Parameter	Description
Trunk Group ID **[TrunkGroupSettings_TrunkGroupId]**	The Trunk Group ID that you want to configure.
Channel Select Mode **[TrunkGroupSettings_ChannelSelectMode]**	The method for which IP-to-Tel calls are assigned to channels pertaining to a Trunk Group. For a detailed description of this parameter, refer to the global parameter ChannelSelectMode. ▪ **[0]** By Dest Phone Number. ▪ **[1]** Cyclic Ascending (default) ▪ **[2]** Ascending ▪ **[3]** Cyclic Descending ▪ **[4]** Descending ▪ **[5]** Dest Number + Cyclic Ascending ▪ **[6]** By Source Phone Number ▪ **[7]** Trunk Cyclic Ascending (applicable only to digital interfaces) ▪ **[8]** Trunk & Channel Cyclic Ascending (applicable only to digital interfaces)
Registration Mode **[TrunkGroupSettings_RegistrationMode]**	Registration method for the Trunk Group: ▪ **[1]** Per Gateway = Single registration for the entire device (default). This mode is applicable only if a default Proxy or Registrar IP are configured, and Registration is enabled (i.e., parameter IsRegisterUsed is set to 1). In this mode, the SIP URI user part in the From, To, and Contact headers is set to the value of the global registration parameter GWRegistrationName or username if GWRegistrationName is not configured. ▪ **[0]** Per Endpoint = Each channel in the Trunk Group registers individually. The registrations are sent to the ServingIPGroupID if defined in the table, otherwise to the default Proxy, and if no default Proxy, then to the Registrar IP. ▪ **[4]** Don't Register = No registrations are sent by endpoints pertaining to the Trunk Group. For example, if the device is configured globally to register all its endpoints (using the parameter ChannelSelectMode), you can exclude some endpoints from being registered by assigning them to a Trunk Group and configuring the Trunk Group registration mode to 'Don't Register'. ▪ **[5]** Per Account = Registrations are sent (or not) to an IP Group, according to the settings in the Account table (refer to "Configuring the Account Table" on page 118). **Notes:** ▪ To enable Trunk Group registrations, configure the global parameter IsRegisterNeeded to 1. This is unnecessary for 'Per Account' registration mode. ▪ If no mode is selected, the registration is performed according to the global registration parameter ChannelSelectMode.

Parameter	Description
	▪ If the device is configured globally (ChannelSelectMode) to register Per Endpoint, and a channels Group comprising four channels is configured to register Per Gateway, the device registers all channels except the first four channels. The channels Group of these four channels sends a single registration request.
Serving IP Group ID **[TrunkGroupSettings_ServingIPGroup]**	The Serving IP Group ID to where INVITE messages initiated by this Trunk Group's endpoints are sent. The actual destination to where these INVITE messages are sent is according to the Proxy Set ID (refer to "Configuring the Proxy Sets Table" on page 123) associated with this Serving IP Group. The Request URI hostname in the INVITE and REGISTER messages (except for 'Per Account' registration modes) is set to the value of the field 'SIP Group Name' defined in the 'IP Group' table (refer to "Configuring the IP Groups" on page 113). If no Serving IP Group ID is selected, the INVITE messages are sent to the default Proxy or according to the 'Outbound IP Routing Table' (refer to "Configuring the Outbound IP Routing Table" on page 158). **Note:** If the parameter PreferRouteTable is set to 1 (refer to "Configuring Proxy and Registration Parameters" on page 122), the routing rules in the 'Outbound IP Routing Table' prevail over the selected Serving IP Group ID.
Gateway Name **[TrunkGroupSettings_GatewayName]**	The host name used in the SIP From header in INVITE messages, and as a host name in From/To headers in REGISTER requests. If not configured, the global parameter SIPGatewayName is used instead.
Contact User **[TrunkGroupSettings_ContactUser]**	The user part in the SIP Contact URI in INVITE messages, and as a user part in From, To, and Contact headers in REGISTER requests. This is applicable only if the field 'Registration Mode' is set to 'Per Account', and the Registration through the Account table is successful. **Notes:** ▪ If registration fails, then the userpart in the INVITE Contact header contains the source party number. ▪ The 'ContactUser' parameter in the 'Account Table' page overrides this parameter.

3.3.5.4 Protocol Definition

The **Protocol Definition** submenu allows you to configure the main SIP protocol parameters. This submenu contains the following page items:

- SIP General Parameters (refer to "SIP General Parameters" on page 108)
- DTMF & Dialing (refer to "DTMF & Dialing Parameters" on page 110)

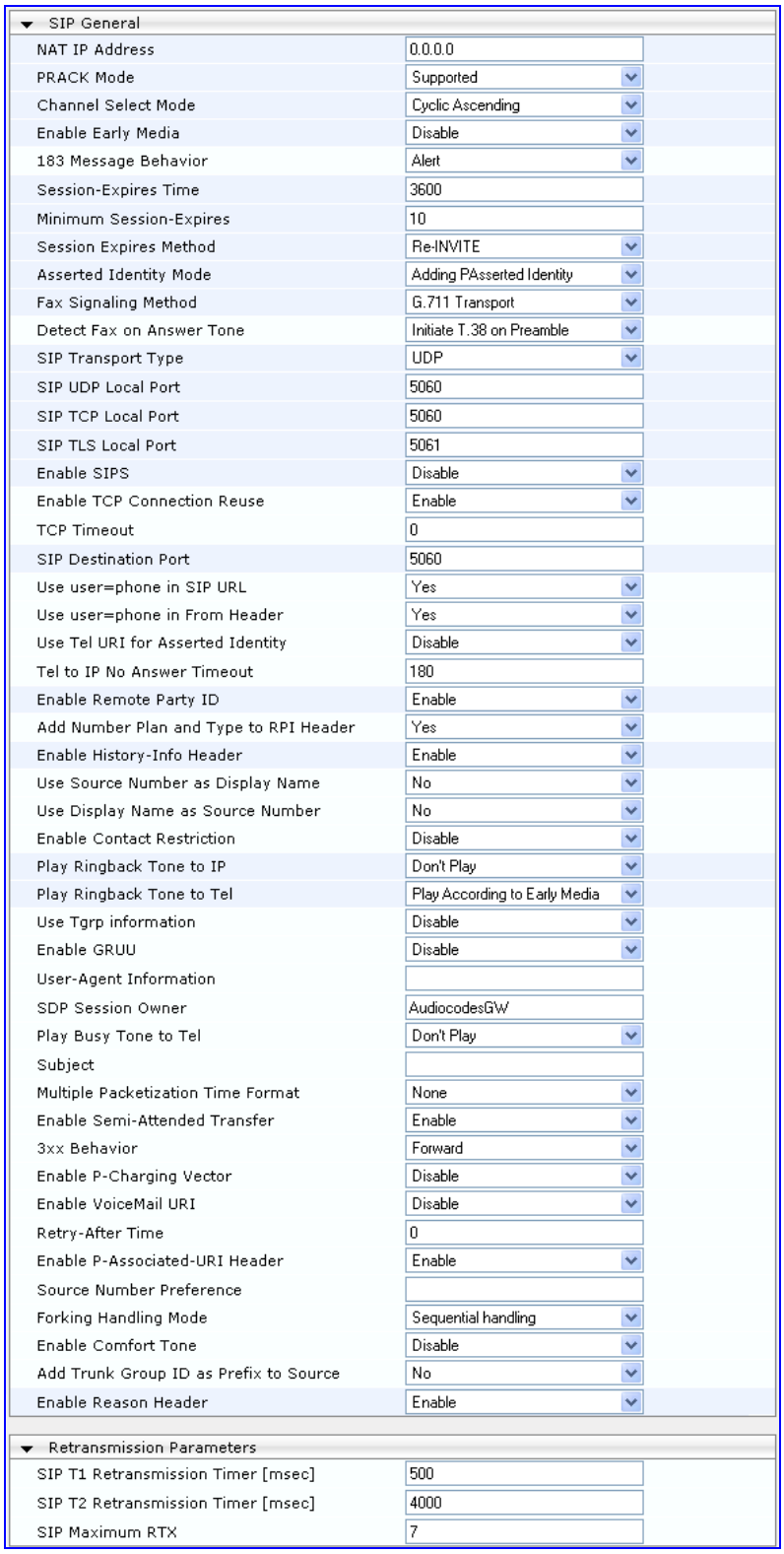
3.3.5.4.1 Configuring SIP General Parameters

The 'SIP General Parameters' page is used to configure general SIP parameters. For a description of the parameters appearing on this page, refer to "Configuration Parameters Reference" on page 247.

➢ **To configure the general SIP protocol parameters:**

1. Open the 'SIP General Parameters' page (**Configuration** tab > **Protocol Configuration** menu > **Protocol Definition** submenu > **SIP General Parameters** page item).

Figure 3-69: SIP General Parameters Page



SIP General	
NAT IP Address	0.0.0.0
PRACK Mode	Supported
Channel Select Mode	Cyclic Ascending
Enable Early Media	Disable
183 Message Behavior	Alert
Session-Expires Time	3600
Minimum Session-Expires	10
Session Expires Method	Re-INVITE
Asserted Identity Mode	Adding PAsserted Identity
Fax Signaling Method	G.711 Transport
Detect Fax on Answer Tone	Initiate T.38 on Preamble
SIP Transport Type	UDP
SIP UDP Local Port	5060
SIP TCP Local Port	5060
SIP TLS Local Port	5061
Enable SIPS	Disable
Enable TCP Connection Reuse	Enable
TCP Timeout	0
SIP Destination Port	5060
Use user=phone in SIP URL	Yes
Use user=phone in From Header	Yes
Use Tel URI for Asserted Identity	Disable
Tel to IP No Answer Timeout	180
Enable Remote Party ID	Enable
Add Number Plan and Type to RPI Header	Yes
Enable History-Info Header	Enable
Use Source Number as Display Name	No
Use Display Name as Source Number	No
Enable Contact Restriction	Disable
Play Ringback Tone to IP	Don't Play
Play Ringback Tone to Tel	Play According to Early Media
Use Tgrp information	Disable
Enable GRUU	Disable
User-Agent Information	
SDP Session Owner	AudiocodesGW
Play Busy Tone to Tel	Don't Play
Subject	
Multiple Packetization Time Format	None
Enable Semi-Attended Transfer	Enable
3xx Behavior	Forward
Enable P-Charging Vector	Disable
Enable VoiceMail URI	Disable
Retry-After Time	0
Enable P-Associated-URI Header	Enable
Source Number Preference	
Forking Handling Mode	Sequential handling
Enable Comfort Tone	Disable
Add Trunk Group ID as Prefix to Source	No
Enable Reason Header	Enable

Retransmission Parameters	
SIP T1 Retransmission Timer [msec]	500
SIP T2 Retransmission Timer [msec]	4000
SIP Maximum RTX	7

2. Configure the parameters as required.
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

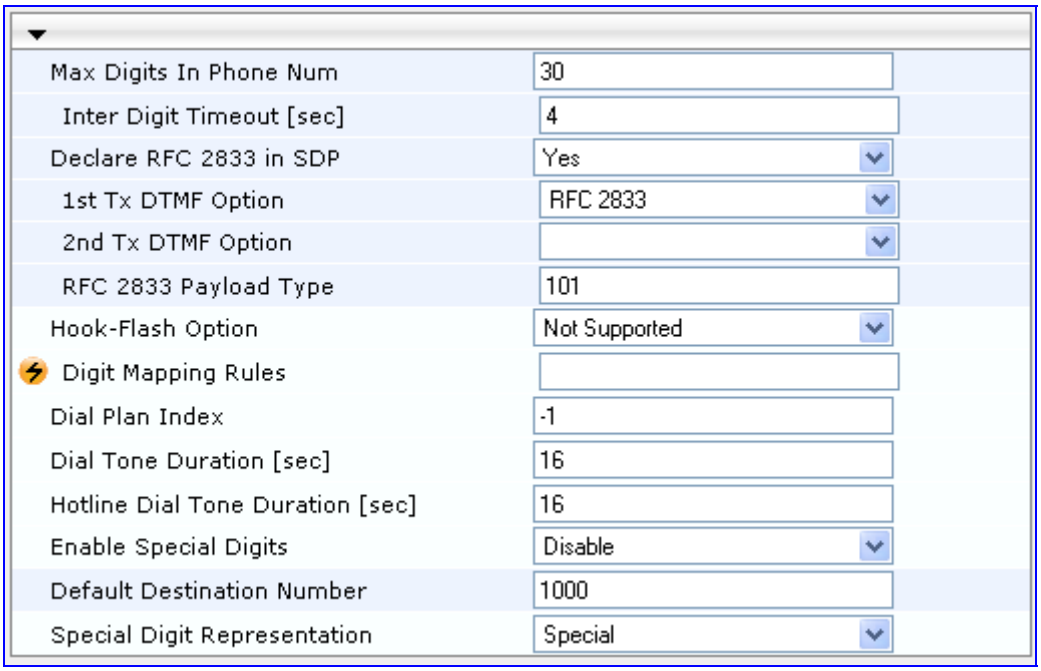
3.3.5.4.2 Configuring DTMF and Dialing Parameters

The 'DTMF & Dialing' page is used to configure parameters associated with dual-tone multi-frequency (DTMF) and dialing. For a description of the parameters appearing on this page, refer to "Configuration Parameters Reference" on page 247.

➢ **To configure the DTMF and dialing parameters:**

1. Open the 'DTMF & Dialing' page (**Configuration** tab > **Protocol Configuration** menu > **Protocol Definition** submenu > **DTMF & Dialing** page item).

Figure 3-70: DTMF & Dialing Page



2. Configure the parameters as required.
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

3.3.5.5 Application Network Setting

The **Application Network Setting** submenu allows you to configure SIP signaling routing domains and interfaces. This submenu contains the following page items:

- SRD Table (refer to "Configuring the Signaling Routing Domain Table" on page 110)
- SIP Interface Table (refer to "Configuring the SIP Interface Table" on page 112)

3.3.5.5.1 Configuring the Signaling Routing Domain Table

The 'SRD Table' page allows you to configure up to five signaling routing domains (SRD). An SRD is a set of definitions of IP interfaces, device resources, SIP behaviors and other definitions that together create (from the IP user's perspective) from one physical device, multiple virtual multi-service gateways.

SRD provides the following:

- Multiple, different SIP signaling (SRD associated with a SIP Interface) and RTP media

(associated with a Media Realm) interfaces for multiple Layer-3 networks.

- Ability to operate with multiple gateway customers that may reside either in the same or in different Layer-3 networks as the device. This allows separation of signaling traffic between different customers. In such a scenario, the device is configured with multiple SRD's.

Typically, one SRD is defined for each group of SIP User Agents/UA (e.g. proxies, IP phones, application servers, gateways, softswitches) that communicate with each other. This provides these entities with VoIP services that reside on the same Layer-3 network (must be able to communicate without traversing NAT devices and must not have overlapping IP addresses).

Routing from one SRD to another is possible, whereby each routing destination (IP Group or destination address) must indicate the SRD to which it belongs.

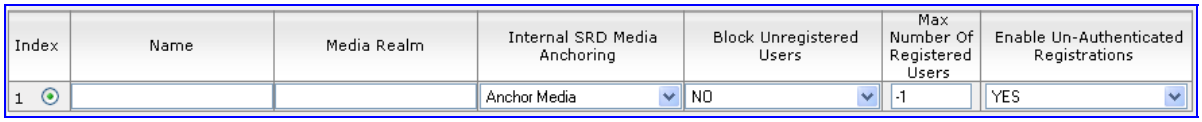
The configuration of an SRD includes assigning it a unique name and assigning it a Media Realm (media port range associated with a Media IP interface, defined in the 'SIP Media Realm' table) as well as associating it with a SIP Signaling interface. Once configured, the SRD can then be assigned to an IP Group (in the IP Group table) and to a Proxy Set (in the Proxy Set table).

Note: The 'SRD' table can also be configured using the *ini* file table parameter SRD (refer to SBC Parameters).

➢ **To configure the SRD table:**

1. Open the 'SRD Table' page (**Configuration** tab > **Protocol Configuration** menu > **Application Network Settings** submenu > **SRD Table**).

Figure 3-71: SRD Table Page



2. Add an entry and then configure it according to the table below.
3. Click the **Apply** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

Table 3-18: SRD Table Parameters

Parameter	Description
Name **[SRD_Name]**	Mandatory descriptive name of the SRD. The valid value can be a string of up to 21 characters.
Media Realm **[SRD_MediaRealm]**	Determines the media ports associated with the specific SRD. This is the name as appears in the 'SIP Media Realm' table (CpMediaRealm). The valid value is a string of up to 40 characters. **Note:** If the Media Realm is later deleted from the 'SIP Media Realm' table, then this name becomes invalid in the SRD table.

3.3.5.5.2 Configuring the SIP Interface Table

The 'SIP Interface Table' page allows you to configure up to six SIP Interfaces. A SIP Interface represents a SIP SIP signaling interface (IPv4), which is a combination of ports (UDP, TCP, and TLS) associated with a specific IP address and an SRD ID. SIP Interfaces allow you to use different SIP signaling interfaces for each of the two legs (i.e., each SIP user agent communicates with a specific SRD).

SIP Interfaces are used for the following:

- Defining different SIP signaling ports (listening UDP, TCP, and TLS, and the UDP source ports) for a single or for multiple interfaces.
- Differentiating between the different applications supported by the device (i.e., SAS or Gateway\IP2IP).
- Separating signaling traffic of different customers to use different routing tables, manipulations, SIP definitions, and so on.

Notes:

- Each SIP Interface must have a unique signaling port (i.e., no two SIP Interfaces can share the same port - no port overlapping).
- Only one signaling interface (SIP Interface) per application type is allowed per SRD.
- The 'SIP Interface' table can also be configured using the *ini* file table parameter SIPInterface.

➢ **To configure the SIP Interface table:**

1. Open the 'SIP Interface Table' page (**Configuration** tab > **Protocol Configuration** menu > **Application Network Settings** submenu > **SIP Interface Table**).

Figure 3-72: SRD Table Page

Index	Network Interface	Application Type	UDP Port	TCP Port	TLS Port	SRD
1	Media1	GW\IP2IP	5060	5060	5061	0
2	Media2	SAS	5080	5080	5081	1

2. Add an entry and then configure it according to the table below.
3. Click the **Apply** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

Table 3-19: SIP Interface Table Parameters

Parameter	Description
Network Interface **[SIPInterface_NetworkInterface]**	Corresponding network interface (IPv4) name, as configured in the 'Multiple Interface' table (InterfaceTable). The default is "Not Configured". **Note:** The value of this parameter must be exactly as configured in the 'Multiple Interface' table ('Interface Name' field).
Application Type **[SIPInterface_ApplicationType]**	Determines the application type associated with the SIP Interface.

Parameter	Description
	▪ **[0]** GW/IP2IP (default) = IP-to-IP routing application and regular gateway functionality ▪ **[1]** SAS = Stand-Alone Survivability (SAS) application
UDP Port **[SIPInterface_UDPPort]**	Determines the listening and source UDP port. The valid range is 1 to 65534. The default is 5060. **Note:** This port must be outside of the RTP port range.
TCP Port **[SIPInterface_TCPPort]**	Determines the listening TCP port. The valid range is 1 to 65534. The default is 5060. **Note:** This port must be outside of the RTP port range.
TLS Port **[SIPInterface_TLSPort]**	Determines the listening TLS port. The valid range is 1 to 65534. The default is 5061. **Note:** This port must be outside of the RTP port range.
SRD **[SIPInterface_SRD]**	Determines the SRD ID (configured in the 'SRD' table) associated with the SIP Interface. The default is 0. **Note:** Each SRD can be associated with up to two SIP Interfaces, where each SIP Interface pertains to a different Application Type (GW/IP2IP and SAS).

3.3.5.6 Proxies, Registration, IP Groups

The **Proxies, Registration, IP Groups** submenu allows you to configure SIP proxy servers, registration parameters, and IP Groups. This submenu includes the following items:

- IP Group Table (refer to "Configuring the IP Groups" on page 113)
- Account Table (refer to "Configuring the Account Table" on page 118)
- Proxy & Registration (refer to "Configuring Proxy and Registration Parameters" on page 122)
- Proxy Sets Table (refer to "Configuring the Proxy Sets Table" on page 123)

3.3.5.6.1 Configuring the IP Groups

The 'IP Group Table' page allows you to create up to nine logical IP entities called *IP Groups*. These IP Groups are used for call routing. The IP Group can be used as a destination entity in the 'Outbound IP Routing Table', and as a Serving IP Group in the 'Trunk Group Settings' (refer to "Configuring Trunk Group Settings" on page 104) and 'Account' (refer to "Configuring the Account Table" on page 118) tables. These call routing tables are used for identifying the IP Group from where the INVITE is sent for obtaining a digest user/password from the 'Account' table if there is a need to authenticate subsequent SIP requests in the call. The IP Group can also be implemented in IP-to-Tel call routing (or inbound IP routing for IP-to-IP routing) as a source IP Group.

The IP Groups can be assigned various entities such as a Proxy Set ID, which represents an IP address (created in "Configuring the Proxy Sets Table" on page 123). You can also assign the IP Group with a host name and other parameters that reflect parameters sent in the SIP Request From\To headers.

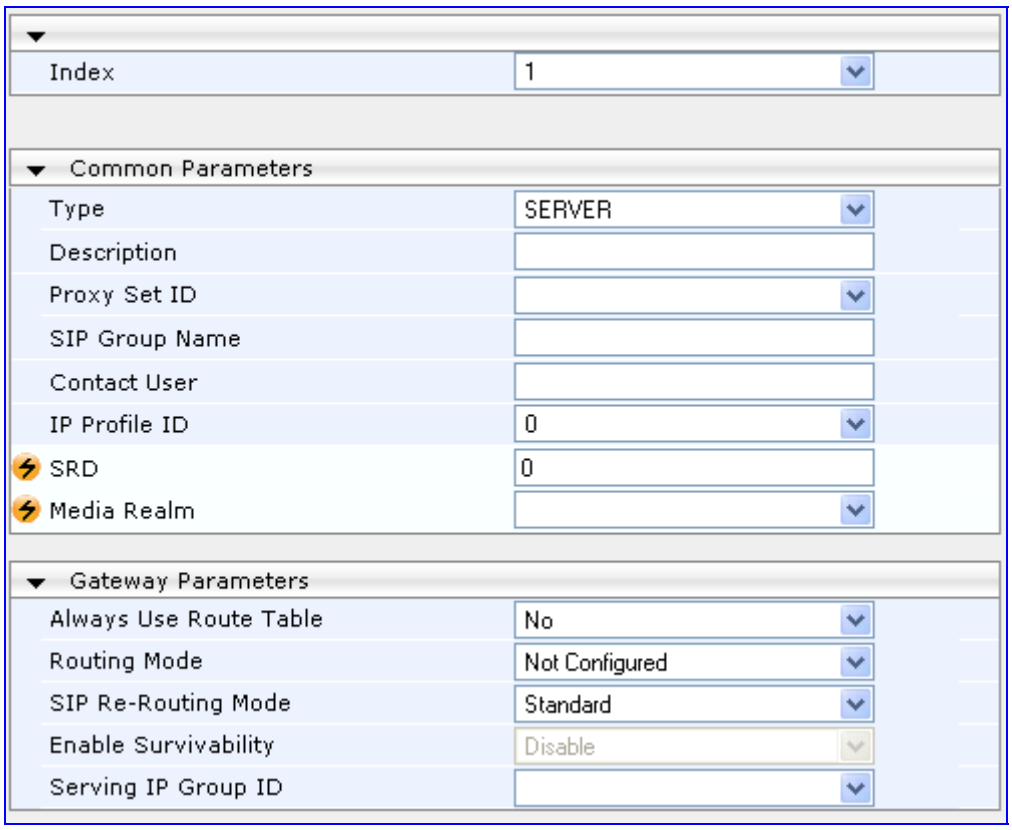
Notes:

- When working with multiple IP Groups, the default Proxy server should not be used (i.e., the parameter IsProxyUsed must be set to 0).
- You can also configure the IP Groups table using the *ini* file table parameter IPGroup (refer to "SIP Configuration Parameters" on page 284).

➢ **To configure IP Groups:**

1. Open the 'IP Group Table' page (**Configuration** tab > **Protocol Configuration** menu > **Proxies, Registration, IP Groups** submenu > **IP Group Table** page item).

Figure 3-73: IP Group Page



2. Configure the IP group parameters according to the table below.
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" **on page 199.**

Table 3-20: IP Group Parameters

Parameter	Description
Common Parameters	
Type **[IPGroup_Type]**	The IP Group can be defined as one of the following types: ▪ SERVER = used when the destination address (configured by the Proxy Set) of the IP Group (e.g., ITSP, Proxy, IP-PBX, or Application server) is known. ▪ USER = represents a group of users (such as IP phones and softphones) where their location is dynamically obtained by the device when REGISTER requests and responses traverse (or are terminated) by the device. These users are considered remote (far-end) users. Typically, this IP Group is configured with a Serving IP Group that represents an IP-PBX, Application or Proxy server that serves this USER-type IP Group. Each SIP request sent by a user of this IP Group is proxied to the Serving IP Group. For registrations, the device updates its internal database with the AOR and contacts of the users. Digest authentication using SIP 401/407 responses (if needed) is performed by the Serving IP Group. The device forwards these responses directly to the SIP users. To route a call to a registered user, a rule must be configured in the 'Outbound IP Routing Table' table (refer to Configuring the Outbound IP Routing Table). The device searches the dynamic database (by using the request URI) for an entry that matches a registered AOR or Contact. Once an entry is found, the IP destination is obtained from this entry, and a SIP request is sent to the destination. The device supports up to 100 registered users. The device also supports NAT traversal for the SIP clients that are behind NAT. In this case, the device must be defined with a global IP address. **Note:** This field is available only if the IP-to-IP application is enabled.
Description **[IPGroup_Description]**	Brief string description of the IP Group. The value range is a string of up to 29 characters. The default is an empty field.
Proxy Set ID **[IPGroup_ProxySetId]**	Selects the Proxy Set ID (defined in "Configuring the Proxy Sets Table" on page 123) to associate with the IP Group. All INVITE messages configured to be 'sent' to the specific IP Group are in fact sent to the IP address associated with this Proxy Set. The range is 1-5. **Notes:** ▪ Proxy Set ID 0 must **not** be selected; this is the device's default Proxy. ▪ The Proxy Set is applicable only to SERVER-type IP Groups.

Parameter	Description
SIP Group Name **[IPGroup_SIPGroupName]**	The request URI host name used in INVITE and REGISTER messages that are sent to this IP Group, or the host name in the From header of INVITE messages received from this IP Group. If not specified, the value of the global parameter ProxyName (refer to "Configuring the Proxy and Registration Parameters" on page 122) is used instead. The value range is a string of up to 49 characters. The default is an empty field. **Note:** If the IP Group is of type USER, this parameter is used internally as a hostname in the request URI for PSTN-to-IP initiated calls. For example, if an incoming call from the device's T1 trunk is routed to a USER-type IP Group, the device first forms the request URI (<destination_number>@<SIP Group Name>), and then it searches the user's internal database for a match.
Contact User **[IPGroup_ContactUser]**	Defines the user part for the From, To, and Contact headers of SIP REGISTER messages, and the user part for the Contact header of INVITE messages that are received from this IP Group and forwarded by the device to another IP Group. **Notes:** ▪ This parameter is applicable only to USER-type IP Groups. ▪ This parameter is overridden by the 'Contact User' parameter in the 'Account' table (refer to "Configuring the Account Table" on page 118).
IP Profile ID **[IPGroup_ProfileId]**	The IP Profile that you want assigned to this IP Group. The default is 0. **Note:** IP Profiles are configured using the parameter IPProfile (refer to "Configuring P Profile Settings" on page 133).
SRD **[IPGroup_SRD]**	The SRD associated with the IP Group. The default is 0.
Media Realm **[IPGroup_MediaRealm]**	The Media Realm name associated with this IP Group. **Note:** Media Realms are configured using the parameter Media Realm table.
Gateway Parameters	
Always Use Route Table **[IPGroup_AlwaysUseRouteTable]**	Determines the Request URI host name in outgoing INVITE messages. ▪ **[0]** Disable (default). ▪ **[1]** Enable = The device uses the IP address (or domain name) defined in the 'Outbound IP Routing Table' (refer to "Configuring the Outbound IP Routing Table" on page 158) as the Request URI host name in outgoing INVITE messages, instead of the value entered in the 'SIP Group Name' field.
Routing Mode **[IPGroup_RoutingMode]**	Defines the routing mode for outgoing SIP INVITE messages. ▪ **[-1]** Not Configured = The routing is according to the selected Serving IP Group. If no Serving IP Group is selected, the device routes the call according to the 'Outbound IP Routing Table' (refer to Configuring the Outbound IP Routing Table). (Default)

Parameter	Description
	▪ **[0]** Routing Table = The device routes the call according to the 'Outbound IP Routing Table'. ▪ **[1]** Serving IP Group = The device sends the SIP INVITE to the selected Serving IP Group. If no Serving IP Group is selected, the default IP Group is used. If the Proxy server(s) associated with the destination IP Group is not alive, the device uses the 'Outbound IP Routing Table' (if the parameter IsFallbackUsed is set 1, i.e., fallback enabled - refer to Configuring Proxy and Registration Parameters on page 122). ▪ **[2]** Request-URI = The device sends the SIP INVITE to the IP address according to the received SIP Request-URI host name. **Note:** This field is available only if the IP-to-IP application is enabled.
SIP Re-Routing Mode **[IPGroup_SIPReRoutingMode]**	Determines the routing mode after a call redirection (i.e., a 3xx SIP response is received) or transfer (i.e., a SIP REFER request is received). ▪ **[0]** Standard = INVITE messages that are generated as a result of Transfer or Redirect are sent directly to the URI, according to the Refer-To header in the REFER message or Contact header in the 3xx response (default). ▪ **[1]** Proxy = Sends a new INVITE to the Proxy. **Note:** Applicable only if a Proxy server is used and the parameter AlwaysSendtoProxy is set to 0. ▪ **[2]** Routing Table = Uses the Routing table to locate the destination and then sends a new INVITE to this destination. **Notes:** ▪ When this parameter is set to **[1]** and the INVITE sent to the Proxy fails, the device re-routes the call according to the Standard mode **[0].** ▪ When this parameter is set to **[2]** and the INVITE fails, the device re-routes the call according to the Standard mode **[0].** If DNS resolution fails, the device attempts to route the call to the Proxy. If routing to the Proxy also fails, the Redirect / Transfer request is rejected. ▪ When this parameter is set to **[2],** the XferPrefix parameter can be used to define different routing rules for redirected calls. ▪ This parameter is ignored if the parameter AlwaysSendToProxy is set to 1.
Enable Survivability **[IPGroup_EnableSurvivability]**	Determines whether Survivability mode is enabled for USER-type IP Groups. ▪ Disable (default). ▪ Enable = Survivability mode is enabled. The device records in its local database the registration messages sent by the clients belonging to the USER-type IP Group. If communication with the Serving IP Group (e.g., IP-PBX) fails, the USER-type IP Group enters into Survivability

Parameter	Description
	mode in which the device uses its database for routing calls between the clients (e.g., IP phones) of the USER-type IP Group. The RTP packets between the IP phones in Survivability mode always traverse through the device. In Survivability mode, the device is capable of receiving new registrations. When the Serving IP Group is available again, the device returns to normal mode, sending INVITE and REGISTER messages to the Serving IP Group. **Notes:** ▪ This field is available only if the IP-to-IP application is enabled. ▪ This parameter is applicable only to USER-type IP Groups.
Serving IP Group ID **[IPGroup_ServingIPGroup]**	If configured, INVITE messages initiated from the IP Group are sent to this Serving IP Group (range 1 to 9). In other words, the INVITEs are sent to the address defined for the Proxy Set associated with this Serving IP Group. The Request URI host name in these INVITE messages are set to the value of the parameter 'SIP Group Name' defined for the Serving IP Group. **Notes:** ▪ This field is available only if the IP-to-IP application is enabled. ▪ If the parameter PreferRouteTable is set to 1, the routing rules in the 'Outbound IP Routing Table' takes precedence over this 'Serving IP Group ID' parameter. ▪ If this parameter is not configured, the INVITE messages are sent to the default Proxy or according to the 'Outbound IP Routing Table'.

3.3.5.6.2 Configuring the Account Table

The 'Account Table' page allows you to define *accounts* per Trunk Group (*Served Trunk Group*) or per source IP Group (Served IP Group) for registration and/or digest authentication (user name and password) to a destination IP address (*Serving IP Group*). The Account table can be used, for example, to register to an Internet Telephony Service Provider (ITSP) on behalf of an IP-PBX to which the device is connected. The registrations are sent to the Proxy Set ID (refer to "Configuring the Proxy Sets Table" on page 123) associated with these Serving IP Groups.

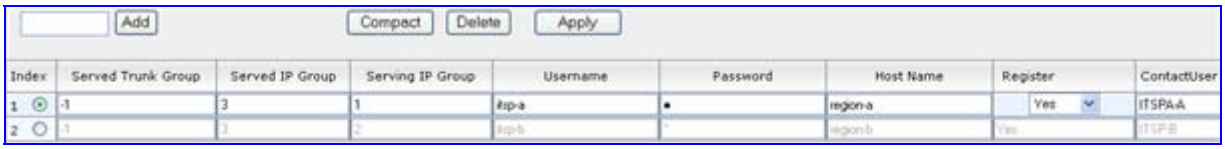
A Trunk Group or source IP Group can register to more than one Serving IP Group (e.g., ITSP's). This can be achieved by configuring multiple entries in the Account table with the same Served Trunk Group or Served IP Group, but with different Serving IP Groups, user name/password, host name, and contact user values.

Note: You can also configure the Account table using the *ini* file table parameter Account (refer to "SIP Configuration Parameters" on page 284).

➢ **To configure Accounts:**

1. Open the 'Account Table' page (**Configuration** tab > **Protocol Configuration** menu > **Proxies, Registration, IP Groups** submenu > **Account Table** page item).

Figure 3-74: Account Table Page



2. To add an Account, in the 'Add' field, enter the desired table row index, and then click **Add**. A new row appears.
3. Configure the Account parameters according to the table below.
4. Click the **Apply** button to save your changes.
5. To save the changes, refer to "Saving Configuration" on page 199.

Note: For a description of the Web interface's table command buttons (e.g., **Duplicate** and **Delete**), refer to "Working with Tables" on page 39.

Table 3-21: Account Table Parameters Description

Parameter	Description
Served Trunk Group **[Account_ServedTrunkGroup]**	The Trunk Group ID for which the device performs registration and/or authentication to a destination IP Group (i.e., Serving IP Group). For Tel-to-IP calls, the Served Trunk Group is the source Trunk Group from where the call initiated. For IP-to-Tel calls, the Served Trunk Group is the 'Trunk Group ID' defined in the 'Inbound IP Routing Table' (refer to "Configuring the Inbound IP Routing Table" on page 163). For defining Trunk Groups, refer to "Configuring the Trunk Group Table" on page 102. **Note:** For the IP2IP application, this parameter must be set to -1 (i.e., no trunk).
Served IP Group **[Account_**ServedIPGroup]	The Source IP Group (e.g., IP-PBX) for which registration and/or authentication is performed. **Note:** This field is applicable only when the IP2IP application is enabled.

Parameter	Description
Serving IP Group **[Account_ServingIPGroup]**	The destination IP Group ID (defined in "Configuring the IP Groups" on page 113) to where the REGISTER requests (if enabled) are sent or Authentication is performed. The actual destination to where the REGISTER requests are sent is the IP address defined for the Proxy Set ID (refer to "Configuring the Proxy Sets Table" on page 123) associated with this IP Group. This occurs only in the following conditions: ▪ The parameter 'Registration Mode' is set to 'Per Account' in the 'Trunk Group Settings' table (refer to "Configuring Trunk Group Settings" on page 104). ▪ The parameter 'Register' in this table is set to 1. In addition, for a SIP call that is identified by both the Served Trunk Group/ Served IP Group and Serving IP Group, the username and password for digest authentication defined in this table is used. For Tel-to-IP calls, the Serving IP Group is the destination IP Group defined in the 'Trunk Group Settings' table or 'Outbound IP Routing Table' (refer to "Configuring the Outbound IP Routing Table" on page 158). For IP-to-Tel calls, the Serving IP Group is the 'Source IP Group ID' defined in the 'Inbound IP Routing Table' (refer to "Configuring the Inbound IP Routing Table" on page 163). **Note:** If no match is found in this table for incoming or outgoing calls, the username and password defined in the 'Authentication' table for FXS interfaces (refer to Configuring Authentication on page 171) or the global parameters (UserName and Password) defined on the 'Proxy & Registration' page.
Username **[Account_Username]**	Digest MD5 Authentication user name (up to 50 characters).
Password **[Account_Password]**	Digest MD5 Authentication password (up to 50 characters). **Note:** After you click the **Apply** button, this password is displayed as an asterisk (*).
Host Name **[Account_HostName]**	Defines the Address of Record (AOR) host name. It appears in REGISTER From/To headers as ContactUser@HostName. For successful registrations, this HostName is also included in the INVITE request's From header URI. If not configured or if registration fails, the 'SIP Group Name' parameter from the 'IP Group' table is used instead. This parameter can be up to 49 characters.

Parameter	Description

Register
[Account_Register]
	Enables registration.
	▪ **[0]** No = Don't register
	▪ **[1]** Yes = Enables registration
	When enabled, the device sends REGISTER requests to the Serving IP Group. In addition, to activate registration, you also need to set the parameter 'Registration Mode' to 'Per Account' in the 'Trunk Group Settings' table for the specific Trunk Group. The Host Name (i.e., host name in SIP From/To headers) and Contact User (user in From/To and Contact headers) are taken from this table upon a successful registration. See the example below:
	`REGISTER sip:xyz SIP/2.0`
	`Via: SIP/2.0/UDP 10.33.37.78;branch=z9hG4bKac1397582418`
	`From: <sip:ContactUser@HostName>;tag=1c1397576231`
	`To: <sip: ContactUser@HostName >`
	`Call-ID: 13975689572612000222560@10.33.37.78`
	`CSeq: 1 REGISTER`
	`Contact: <sip:ContactUser@10.33.37.78>;expires=3600`
	`Expires: 3600`
	`User-Agent: Sip-Gateway/v.6.00A.008.002`
	`Content-Length: 0`
	Notes:
	▪ The Trunk Group account registration is not affected by the parameter IsRegisterNeeded.
	▪ For the IP2IP application, you can configure this table so that a specific IP Group can register to multiple ITSP's.This is performed by defining several rows in this table containing the same Served IP Group, but with different Serving IP Groups, user/password, Host Name and Contact User parameters.
	▪ If registration to an IP Group(s) fails for **all** the accounts defined in this table for a specific Trunk Group, and if this Trunk Group includes all the channels in the Trunk Group, the Trunk Group is set to Out-Of-Service if the parameter OOSOnRegistrationFail is set to 1 (refer to "Proxy & Registration Parameters" on page 122).

Contact User **[Account_ContactUser]**	Defines the AOR user name. It appears in REGISTER From/To headers as ContactUser@HostName, and in INVITE/200 OK Contact headers as ContactUser@<device's IP address>. If not configured, the 'Contact User' parameter from the 'IP Group Table' page is used instead. **Note:** If registration fails, then the user part in the INVITE Contact header contains the source party number.
Application Type **[Account_ApplicationType]**	**Note:** This parameter is not applicable.

3.3.5.6.3 Configuring Proxy and Registration Parameters

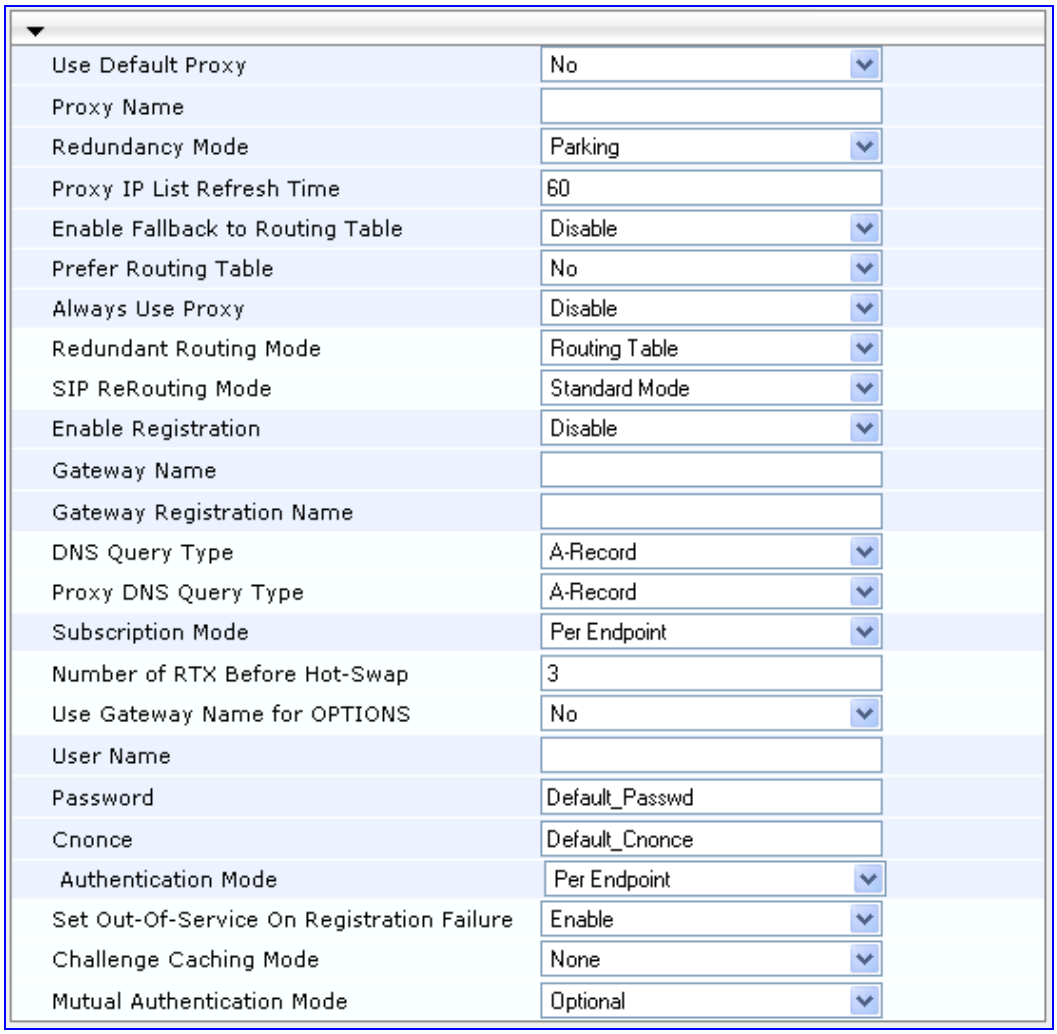
The 'Proxy & Registration' page allows you to configure parameters that are associated with Proxy and Registration. For a description of the parameters appearing on this page, refer to "Configuration Parameters Reference" on page 247.

Note: To view whether the device or its endpoints have registered to a SIP Registrar/Proxy server, refer to "Registration Status" on page 217.

➢ **To configure the Proxy & Registration parameters:**

1. Open the 'Proxy & Registration' page (**Configuration** tab > **Protocol Configuration** menu > **Proxies, Registration, IP Groups** submenu > **Proxy & Registration** page item).

Figure 3-75: Proxy & Registration Page



2. Configure the parameters as required.
3. Click the **Submit** button to save your changes, or click the **Register** or **Un-Register** buttons to save your changes and register / unregister to a Proxy / Registrar.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

Click the **Proxy Set Table** button to open the 'Proxy Sets Table' page to configure groups of proxy addresses. Alternatively, you can open this page from the **Proxy Sets Table** page item (refer to "Configuring the Proxy Sets Table" on page 123 for a description of this page).

3.3.5.6.4 Configuring the Proxy Sets Table

The 'Proxy Sets Table' page allows you to define *Proxy Sets*. A Proxy Set is a group of Proxy servers defined by IP address or fully qualified domain name (FQDN). You can define up to ten Proxy Sets, each having a unique ID number and each containing up to five Proxy server addresses. For each Proxy server address you can define the transport type (i.e., UDP, TCP, or TLS). In addition, Proxy load balancing and redundancy mechanisms can be applied per Proxy Set (if a Proxy Set contains more than one Proxy address).

Proxy Sets can later be assigned to IP Groups of type SERVER only (refer to "Configuring the IP Groups" on page 113). When the device sends an INVITE message to an IP Group, it is sent to the IP address or domain name defined for the Proxy Set that is associated with the specific IP Group. In other words, the Proxy Set represents the **destination** of the call. Typically, for IP-to-IP call routing, at least two Proxy Sets are defined for call destination – one for each leg (IP Group) of the call (i.e., both directions). For example, one Proxy Set for the Internet Telephony Service provider (ITSP) interfacing with one 'leg' of the device and another Proxy Set for the second SIP entity (e.g., ITSP) interfacing with the other 'leg' of the device.

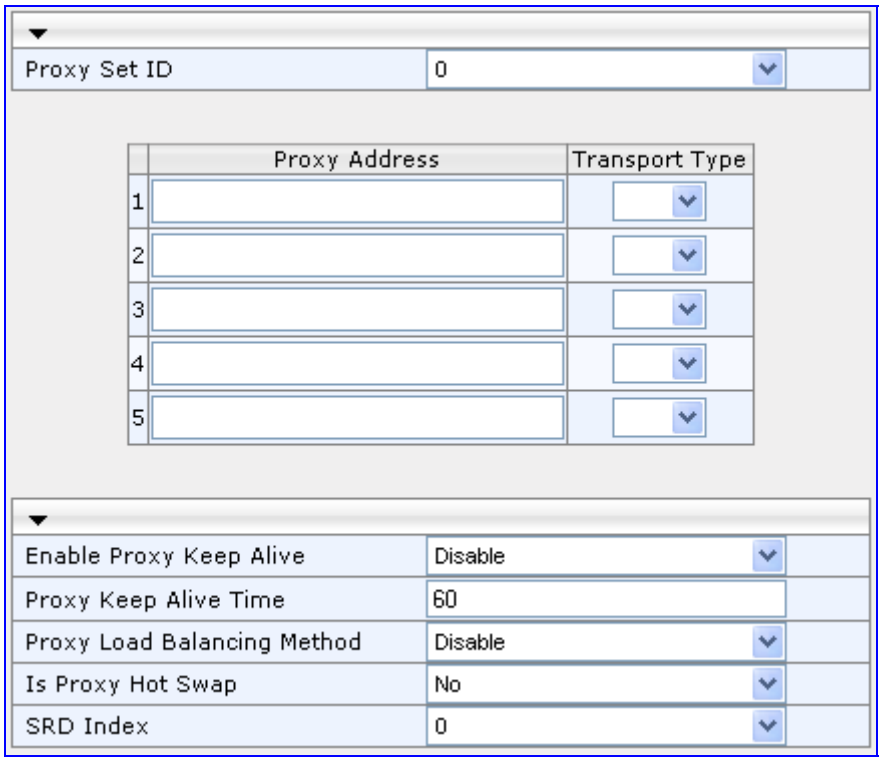
Notes:

- You can also configure the Proxy Sets table using two complementary *ini* file table parameters (refer to "SIP Configuration Parameters" on page 284):
  - ProxyIP: used for creating a Proxy Set ID defined with IP addresses.
  - ProxySet: used for defining various attributes for the Proxy Set ID.
- Proxy Sets can be assigned only to SERVER-type IP Groups.

➢ **To add Proxy servers and configure Proxy parameters:**

1. Open the 'Proxy Sets Table' page (**Configuration** tab > **Protocol Configuration** menu > **Proxies, Registration, IP Groups** submenu > **Proxy Sets Table** page item).

Figure 3-76: Proxy Sets Table Page



2. From the Proxy Set ID drop-down list, select an ID for the desired group.
3. Configure the Proxy parameters according to the following table.
4. Click the **Submit** button to save your changes.
5. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

Table 3-22: Proxy Sets Table Parameters

Parameter	Description
Web: Proxy Set ID EMS: Index **[ProxySet_Index]**	The Proxy Set identification number. The valid range is 0 to 9 (i.e., up to ten Proxy Set ID's can be configured). The Proxy Set ID 0 is used as the default Proxy Set. **Note:** Although not recommended, you can use both default Proxy Set (ID 0) and IP Groups for call routing. For example, on the 'Trunk Group Settings' page (refer to "Configuring Trunk Group Settings" on page 104) you can configure a Serving IP Group to where you want to route specific Trunk Group's channels, while all other device channels use the default Proxy Set. At the same, you can also use IP Groups in the 'Outbound IP Routing Table' (refer to "Configuring the Outbound IP Routing Table" on page 158) to configure the default Proxy Set if the parameter PreferRouteTable is set to 1. To summarize, if the default Proxy Set is used, the INVITE message is sent according to the following preferences: ▪ To the Trunk Group's Serving IP Group ID, as defined in the

Parameter	Description
	'Trunk Group Settings' table. ▪ According to the 'Outbound IP Routing Table' if the parameter PreferRouteTable is set to 1. ▪ To the default Proxy. Typically, when IP Groups are used, there is no need to use the default Proxy, and all routing and registration rules can be configured using IP Groups and the Account tables (refer to "Configuring the Account Table" on page 118).
Proxy Address **[ProxyIp_IpAddress]**	The IP address (and optionally port number) of the Proxy server. Up to five IP addresses can be configured per Proxy Set. Enter the IP address as an FQDN or in dotted-decimal notation (e.g., 201.10.8.1). You can also specify the selected port in the format: <IP address>:<port>. If you enable Proxy Redundancy (by setting the parameter EnableProxyKeepAlive to 1 or 2), the device can operate with multiple Proxy servers. If there is no response from the first (*primary*) Proxy defined in the list, the device attempts to communicate with the other (*redundant*) Proxies in the list. When a redundant Proxy is located, the device either continues operating with it until the next failure occurs, or reverts to the primary Proxy (refer to the parameter ProxyRedundancyMode). If none of the Proxy servers respond, the device goes over the list again. The device also provides real-time switching (Hot-Swap mode) between the primary and redundant proxies (refer to the parameter IsProxyHotSwap). If the first Proxy doesn't respond to the INVITE message, the same INVITE message is immediately sent to the next Proxy in the list. The same logic applies to REGISTER messages (if RegistrarIP is not defined). **Notes:** ▪ If EnableProxyKeepAlive is set to 1 or 2, the device monitors the connection with the Proxies by using keep-alive messages (OPTIONS or REGISTER). ▪ To use Proxy Redundancy, you must specify one or more redundant Proxies. ▪ When a port number is specified (e.g., domain.com:5080), DNS NAPTR/SRV queries aren't performed, even if ProxyDNSQueryType is set to 1 or 2.
Transport Type **[ProxyIp_TransportType]**	The transport type per Proxy server. ▪ **[0]** UDP ▪ **[1]** TCP ▪ **[2]** TLS ▪ **[-1]** = Undefined **Note:** If no transport type is selected, the value of the global parameter SIPTransportType is used (refer to "Configuring SIP General Parameters" on page 108).

Parameter	Description
Web: Proxy Load Balancing Method EMS: Load Balancing Method **[ProxyLoadBalancingMethod]**	Enables the Proxy Load Balancing mechanism per Proxy Set ID. ▪ **[0]** Disable = Load Balancing is disabled (default). ▪ **[1]** Round Robin = Round Robin. ▪ **[2]** Random Weights = Random Weights. When the Round Robin algorithm is used, a list of all possible Proxy IP addresses is compiled. This list includes all IP addresses per Proxy Set, after necessary DNS resolutions (including NAPTR and SRV, if configured). After this list is compiled, the Proxy Keep-Alive mechanism (according to parameters EnableProxyKeepAlive and ProxyKeepAliveTime) tags each entry as 'offline' or 'online'. Load balancing is only performed on Proxy servers that are tagged as 'online'. All outgoing messages are equally distributed across the list of IP addresses. REGISTER messages are also distributed unless a RegistrarIP is configured. The IP addresses list is refreshed according to ProxyIPListRefreshTime. If a change in the order of the entries in the list occurs, all load statistics are erased and balancing starts over again. When the Random Weights algorithm is used, the outgoing requests are not distributed equally among the Proxies. The weights are received from the DNS server by using SRV records. The device sends the requests in such a fashion that each Proxy receives a percentage of the requests according to its' assigned weight. A single FQDN should be configured as a Proxy IP address. The Random Weights Load Balancing is not used in the following scenarios: ▪ The Proxy Set includes more than one Proxy IP address. ▪ The only Proxy defined is an IP address and not an FQDN. ▪ SRV is not enabled (DNSQueryType). ▪ The SRV response includes several records with a different Priority value.
Web/EMS: Enable Proxy Keep Alive **[EnableProxyKeepAlive]**	Determines whether Keep-Alive with the Proxy is enabled or disabled. This parameter is configured per Proxy Set. ▪ **[0]** Disable = Disable (default). ▪ **[1]** Using OPTIONS = Enables Keep-Alive with Proxy using OPTIONS. ▪ **[2]** Using REGISTER = Enable Keep-Alive with Proxy using REGISTER. If set to 'Using OPTIONS', the SIP OPTIONS message is sent every user-defined interval, as configured by the parameter ProxyKeepAliveTime. If set to 'Using REGISTER', the SIP REGISTER message is sent every user-defined interval, as configured by the parameter RegistrationTime. Any response from the Proxy, either success (200 OK) or failure (4xx response) is considered as if the Proxy is communicating correctly. **Notes:** ▪ For Survivability mode for USER-type IP Groups, this parameter must be enabled (1 or 2). ▪ This parameter must be set to 'Using OPTIONS' when Proxy

Parameter	Description
	redundancy is used. ▪ When this parameter is set to 'Using REGISTER', the homing redundancy mode is disabled. ▪ When the active proxy doesn't respond to INVITE messages sent by the device, the proxy is tagged as 'offline'. The behavior is similar to a Keep-Alive (OPTIONS or REGISTER) failure.
Web: Proxy Keep Alive Time EMS: Keep Alive Time **[ProxyKeepAliveTime]**	Defines the Proxy keep-alive time interval (in seconds) between Keep-Alive messages. This parameter is configured per Proxy Set. The valid range is 5 to 2,000,000. The default value is 60. **Note:** This parameter is applicable only if the parameter EnableProxyKeepAlive is set to 1 (OPTIONS). When the parameter EnableProxyKeepAlive is set to 2 (REGISTER), the time interval between Keep-Alive messages is determined by the parameter RegistrationTime.
Web/EMS: Is Proxy Hot-Swap **[IsProxyHotSwap]**	Enables the Proxy Hot-Swap redundancy mode per Proxy Set. ▪ **[0]** No = Disabled (default). ▪ **[1]** Yes = Proxy Hot-Swap mode is enabled. If Proxy Hot-Swap is enabled, the SIP INVITE/REGISTER message is initially sent to the first Proxy/Registrar server. If there is no response from the first Proxy/Registrar server after a specific number of retransmissions (configured by the parameter HotSwapRtx), the INVITE/REGISTER message is resent to the next redundant Proxy/Registrar server.
SRD Index **[ProxySet_**SRD]	The SRD associated with the Proxy Set ID. **Note:** If no SRD is defined for this parameter, by default, SRD ID 0 is associated with the Proxy Set.

3.3.5.7 Coders and Profile Definitions

The **Coders And Profile Definitions** submenu includes the following page items:

- Coders (refer to "Configuring Coders" on page 128)
- Coder Group Settings (refer to "Configuring Coder Groups" on page 130)
- Tel Profile Settings (refer to Configuring Tel Profiles on page 131)
- IP Profile Settings (refer to Configuring IP Profiles on page 133)

Implementing the device's Profile features, provides the device with high-level adaptation when connected to a variety of equipment (at both Tel and IP sides) and protocols, each of which requires different system behavior.

You can assign different Profiles (behavior) per call, using the call routing tables:

- 'Outbound IP Routing Table' page (refer to Configuring the Outbound IP Routing Table on page 158)
- 'Inbound IP Routing Table' page (refer to Configuring the Inbound IP Routing Table on page 163)

In addition, you can associate different Profiles per the device's channels.

Each Profile contains a set of parameters such as coders, T.38 Relay, Voice and DTMF Gain, Silence Suppression, Echo Canceler, RTP DiffServ, Current Disconnect and more. The Profiles feature allows you to customize these parameters or turn them on or off, per source or destination routing and/or per the device's trunks (channels). For example, specific E1/T1 spans can be assigned a Profile that always uses G.711.

Each call can be associated with one or two Profiles - Tel Profile and/or IP Profile. If both IP and Tel profiles apply to the same call, the coders and other common parameters of the preferred Profile (determined by the Preference option) are applied to that call. If the Preference of the Tel and IP Profiles is identical, the Tel Profile parameters take precedence.

Notes:

- The default values of the parameters in the 'Tel Profile Settings' and 'IP Profile Settings' pages are identical to their default values in their respective primary configuration page.
- If you modify a parameter in its primary configuration page (or *ini* file) that also appears in the profile pages, the parameter's new value is automatically updated in the profile pages. However, once you modify any parameter in the profile pages, modifications to parameters in the primary configuration pages (or *ini* file) no longer impact that profile pages.

3.3.5.7.1 Configuring Coders

The 'Coders' page allows you to configure up to ten coders (and their attributes) for the device. The first coder in the list has the highest priority and is used by the device whenever possible. If the far-end device cannot use the first coder, the device attempts to use the next coder in the list, and so on.

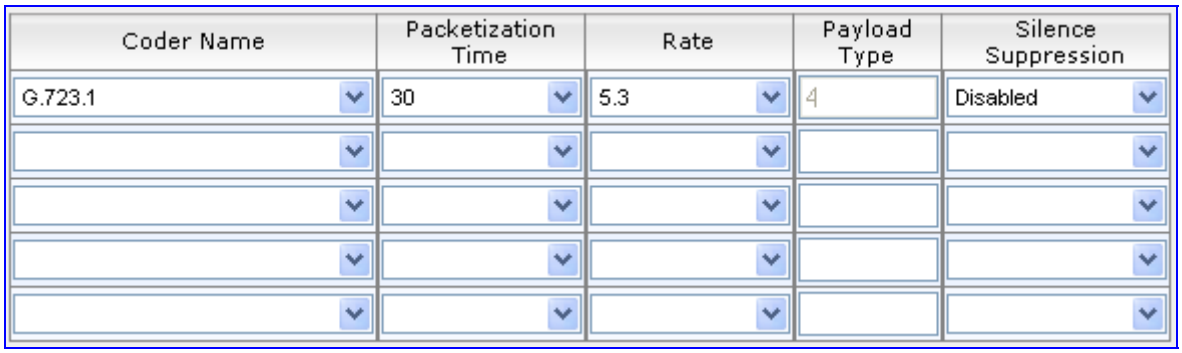
Notes:

- For a list of supported coders and for configuring coders using the *ini* file, refer to the *ini* file parameter table CodersGroup, described in "SIP Configuration Parameters" on page 284.
- For defining groups of coders (which can be assigned to Tel and IP Profiles), refer to "Configuring Coder Groups" on page 130.
- The device always uses the packetization time requested by the remote side for sending RTP packets.
- For an explanation on V.152 support (and implementation of T.38 and VBD coders), refer to "Supporting V.152 Implementation" on page 563.

➢ **To configure the device's coders:**

1. Open the 'Coders' page (**Configuration** tab > **Protocol Configuration** menu > **Coders And Profile Definitions** submenu > **Coders** page item).

Figure 3-77: Coders Page



2. From the 'Coder Name' drop-down list, select the required coder.
3. From the 'Packetization Time' drop-down list, select the packetization time (in msec) for the selected coder. The packetization time determines how many coder payloads are combined into a single RTP packet.
4. From the 'Rate' drop-down list, select the bit rate (in kbps) for the selected coder.
5. In the 'Payload Type' field, if the payload type (i.e., format of the RTP payload) for the selected coder is dynamic, enter a value from 0 to 120 (payload types of 'well-known' coders cannot be modified).
6. From the 'Silence Suppression' drop-down list, enable or disable the silence suppression option for the selected coder.
7. Repeat steps 2 through 6 for the next optional coders.
8. Click the **Submit** button to save your changes.
9. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

Notes:

- A coder (i.e., 'Coder Name') can appear only once in the table.
- If packetization time and/or rate are not specified, the default value is applied.
- Only the packetization time of the first coder in the coder list is declared in INVITE/200 OK SDP, even if multiple coders are defined.
- For G.729, it's also possible to select silence suppression without adaptations.
- If the coder G.729 is selected and silence suppression is disabled (for this coder), the device includes the string 'annexb=no' in the SDP of the relevant SIP messages. If silence suppression is enabled or set to 'Enable w/o Adaptations', 'annexb=yes' is included. An exception to this logic is when the remote gateway is a Cisco device (IsCiscoSCEMode).

3.3.5.7.2 Configuring Coder Groups

The 'Coder Group Settings' page provides a table for defining up to four different coder groups. These coder groups are used in the 'Tel Profile Settings' and 'IP Profile Settings' pages to assign different coders to Profiles. For each coder group, you can define up to ten coders, where the first coder (and its attributes) in the table takes precedence over the second coder, and so on. The first coder is the highest priority coder and is used by the device whenever possible. If the far end device cannot use the coder assigned as the first coder, the device attempts to use the next coder and so on.

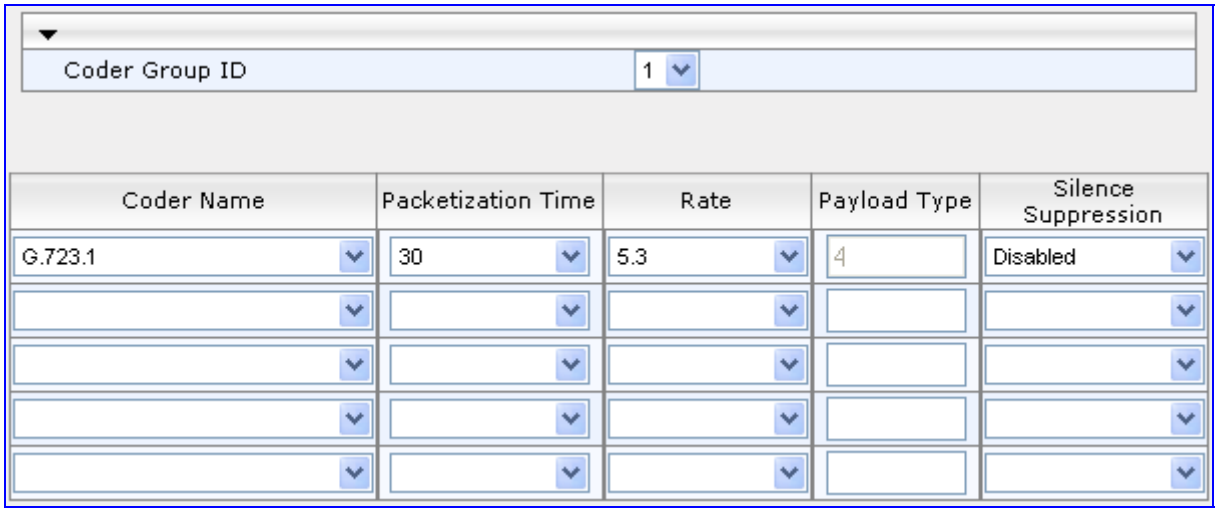
Notes:

- For a list of supported coders and for configuring coders using the *ini* file, refer to the *ini* file parameter table CodersGroup, described in "SIP Configuration Parameters" on page 284.
- Each coder type can appear only once per Coder Group.
- The device always uses the packetization time requested by the remote side for sending RTP packets. If not specified, the packetization time (ptime) is assigned the default value.
- Only the packetization time of the first coder in the defined coder list is declared in INVITE / 200 OK SDP, even if multiple coders are defined.
- For G.729, you can also select silence suppression without adaptations.
- If silence suppression is enabled for G.729, the device includes the string 'annexb=no' in the SDP of the relevant SIP messages. If silence suppression is set to 'Enable w/o Adaptations', 'annexb=yes' is included. An exception is when the remote device is a Cisco gateway (IsCiscoSCEMode).

➢ **To configure coder groups:**

1. Open the 'Coder Group Settings' page (**Configuration** tab > **Protocol Configuration** menu > **Coders And Profile Definitions** submenu > **Coder Group Settings** page item).

Figure 3-78: Coder Group Settings Page



2. From the 'Coder Group ID' drop-down list, select a coder group ID.
3. From the 'Coder Name' drop-down list, select the first coder for the coder group.

4. From the 'Packetization Time' drop-down list, select the packetization time (in msec) for the coder. The packetization time determines how many coder payloads are combined into a single RTP packet.
5. From the 'Rate' drop-down list, select the bit rate (in kbps) for the coder you selected.
6. In the 'Payload Type' field, if the payload type (i.e., format of the RTP payload) for the coder you selected is dynamic, enter a value from 0 to 120 (payload types of 'well-known' coders cannot be modified).
7. From the 'Silence Suppression' drop-down list, enable or disable the silence suppression option for the coder you selected.
8. Repeat steps 3 through 7 for the next coders (optional).
9. Repeat steps 2 through 8 for the next coder group (optional).
10. Click the **Submit** button to save your changes.
11. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

3.3.5.7.3 Configuring Tel Profile

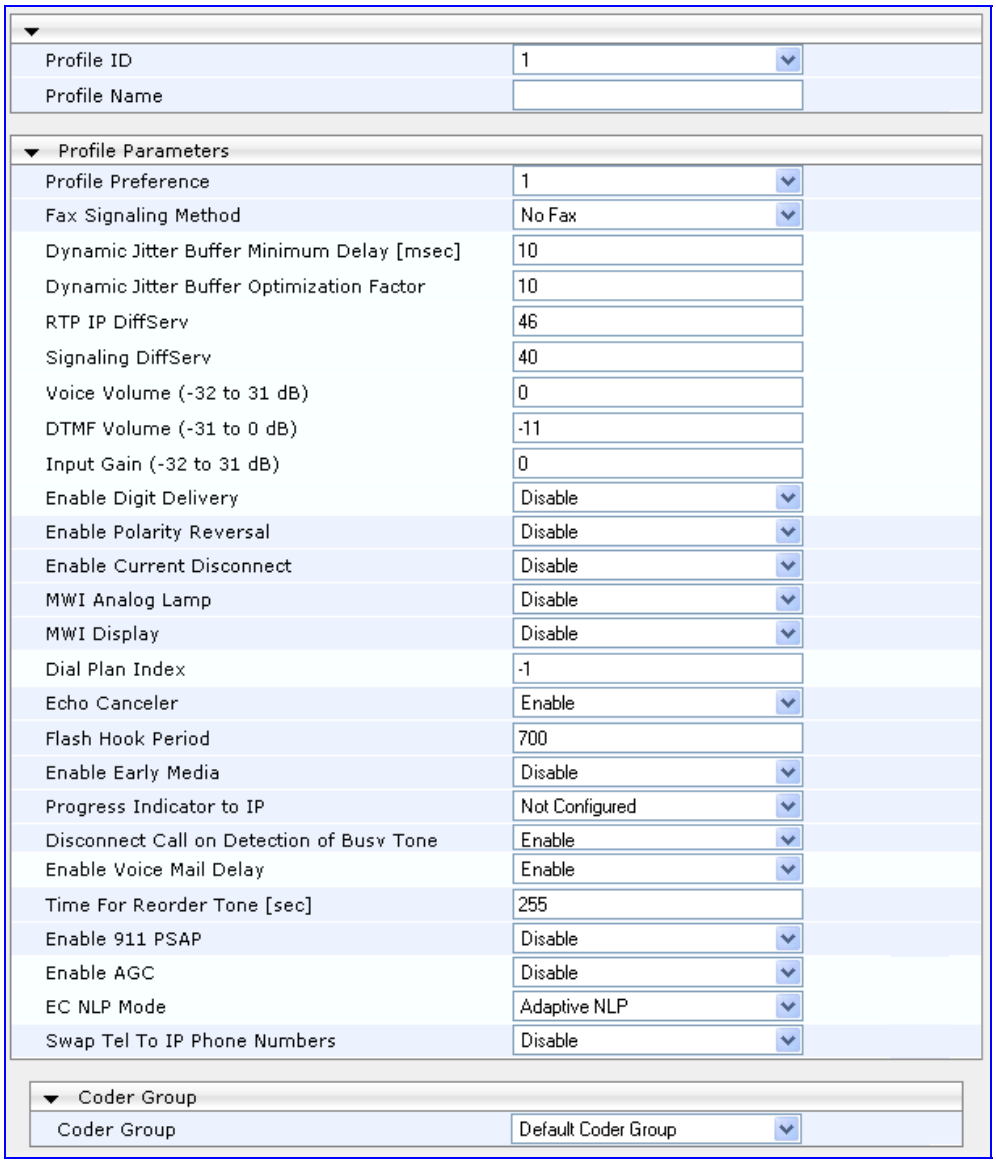
The 'Tel Profile Settings' page allows you to define up to nine Tel Profiles. You can then assign these Tel Profiles to the device's channels (in the 'Trunk Group Table' page), thereby applying different behaviors to different channels.

Note: You can also configure Tel Profiles using the *ini* file table parameter TelProfile (refer to "SIP Configuration Parameters" on page 284).

➢ **To configure Tel Profiles:**

1. Open the 'Tel Profile Settings' page (**Configuration** tab > **Protocol Configuration** menu > **Coders And Profile Definitions** submenu > **Tel Profile Settings** page item).

Figure 3-79: Tel Profile Settings Page



2. From the 'Profile ID' drop-down list, select the Tel Profile identification number you want to configure.
3. In the 'Profile Name' field, enter an arbitrary name that enables you to easily identify the Tel Profile.
4. From the 'Profile Preference' drop-down list, select the priority of the Tel Profile, where '1' is the lowest priority and '20' is the highest. If both IP and Tel profiles apply to the same call, the coders and other common parameters (noted by an asterisk in the description of the parameter TelProfile) of the preferred Profile are applied to that call. If the Preference of the Tel and IP Profiles is identical, the Tel Profile parameters are applied.
   Note: If the coder lists of both IP and Tel Profiles apply to the same call, only the coders common to both are used. The order of the coders is determined by the preference.
5. Configure the Profile's parameters according to your requirements. For detailed information on each parameter, refer to its description on the page in which it is configured as an individual parameter.
6. From the 'Coder Group' drop-down list, select the Coder Group (refer to "Configuring Coder Groups" on page 130) or the device's default coder (refer to "Configuring Coders" on page 128) to which you want to assign the Profile.
7. Repeat steps 2 through 6 to configure additional Tel Profiles (optional).
8. Click the **Submit** button to save your changes.
9. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

3.3.5.7.4 Configuring IP Profiles

The 'IP Profile Settings' page allows you to define up to nine different IP Profiles. You can later assign IP Profiles to routing rules in the call routing tables:

- 'Outbound IP Routing Table' page (refer to "Configuring Outbound IP Routing Table" on page 158)
- 'Inbound IP Routing Table' page (refer to "Configuring the Inbound IP Routing Table" on page 163)

The 'IP Profile Settings' page conveniently groups the different parameters according to application to which they pertain:

- Common Parameters: parameters common to all application types
- Gateway Parameters: parameters applicable to gateway functionality

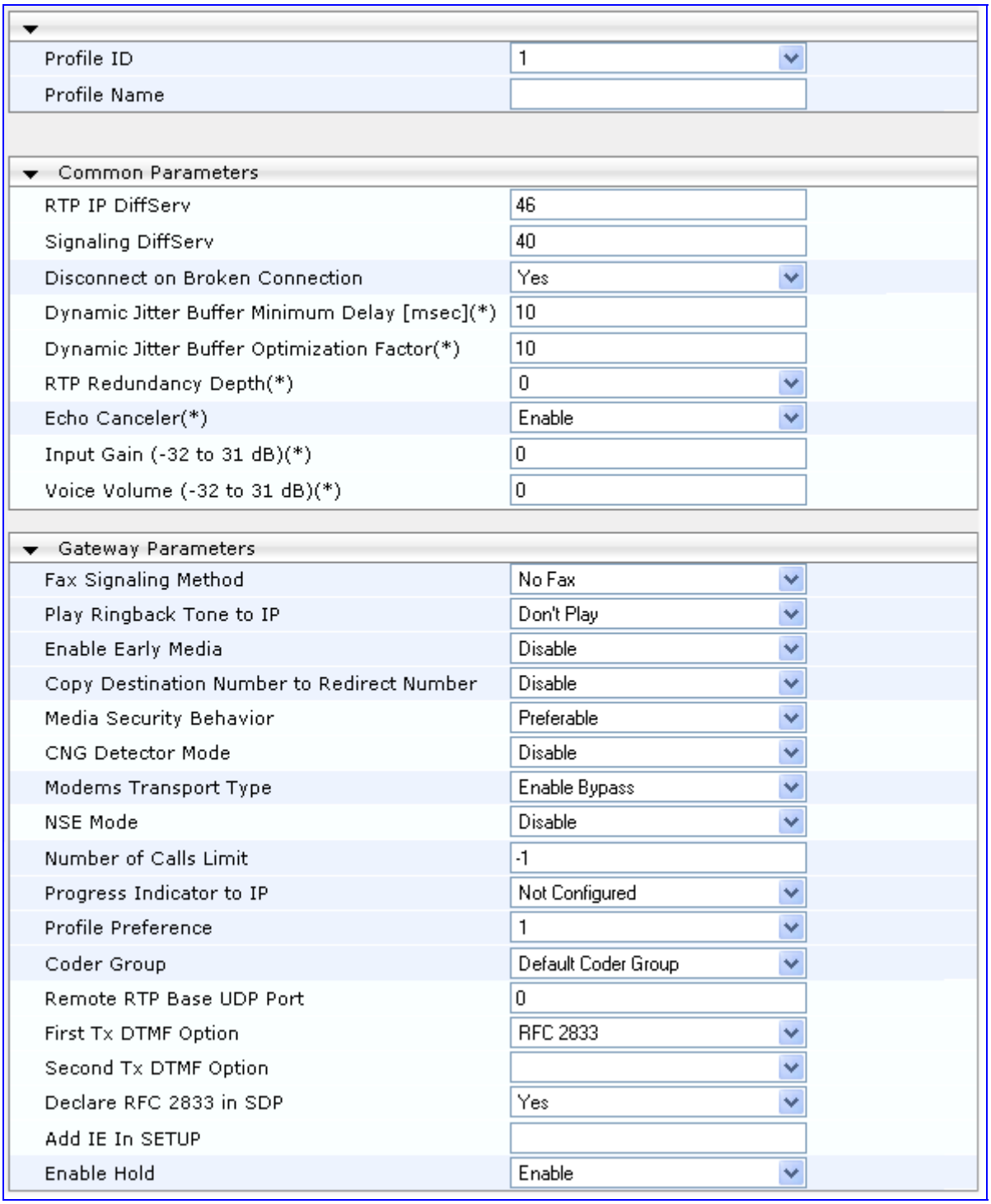
Notes:

- For a detailed description of each parameter in the 'IP Profile' table, refer to its corresponding "global" parameter (configured as an individual parameter).
- IP Profiles can also be implemented when operating with a Proxy server (when the parameter AlwaysUseRouteTable is set to 1).
- You can also configure the IP Profiles using the *ini* file table parameter IPProfile (refer to "SIP Configuration Parameters" on page 284).

➢ **To configure the IP Profile settings:**

1. Open the 'IP Profile Settings' page (**Configuration** tab > **Protocol Configuration** menu > **Coders And Profile Definitions** submenu > **IP Profile Settings**).

Figure 3-80: IP Profile Settings Page



2. From the 'Profile ID' drop-down list, select an identification number for the IP Profile.
3. In the 'Profile Name' field, enter an arbitrary name that allows you to easily identify the IP Profile.

4. From the 'Profile Preference' drop-down list, select the priority of the IP Profile, where '1' is the lowest priority and '20' is the highest. If both IP and Tel profiles apply to the same call, the coders and other common parameters (noted by an asterisk) of the preferred Profile are applied to that call. If the Preference of the Tel and IP Profiles is identical, the Tel Profile parameters are applied.
   Note: If the coder lists of both IP and Tel Profiles apply to the same call, only the coders common to both are used. The order of the coders is determined by the preference.
5. Configure the IP Profile's parameters according to your requirements. Parameters that are unique to IP Profile are described in the table below.
6. From the 'Coder Group' drop-down list, select the coder group that you want to assign to the IP Profile. You can select the device's default coders (refer to "Configuring Coders" on page 128), or one of the coder groups you defined in the 'Coder Group Settings' page (refer to "Configuring Coder Groups" on page 130).
7. Repeat steps 2 through 6 for the next IP Profiles (optional).
8. Click the **Submit** button to save your changes.
9. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

Table 3-23: Description of Parameter Unique to IP Profile

Parameter	Description
Number of Calls Limit	Maximum number of concurrent calls. If the profile is set to some limit, the device maintains the number of concurrent calls (incoming and outgoing) pertaining to the specific profile. A limit value of '-1' indicates that there is no limitation on calls for that specific profile (default). A limit value of '0' indicates that all calls are rejected. When the number of concurrent calls is equal to the limit, the device rejects any new incoming and outgoing calls belonging to that profile.

3.3.5.8 SIP Advanced Parameters

The **SIP Advanced Parameters** submenu allows you to configure advanced SIP control protocol parameters. This submenu contains the following page items:

- Advanced Parameters (refer to "Configuring Advanced Parameters" on page 135)
- Supplementary Services (refer to "Configuring Supplementary Services" on page 138)
- Metering Tones (refer to Configuring Metering Tones on page 139)
- Charge Codes (refer to Configuring the Charge Codes Table on page 139)
- Keypad Features (refer to Configuring Keypad Features on page 141)

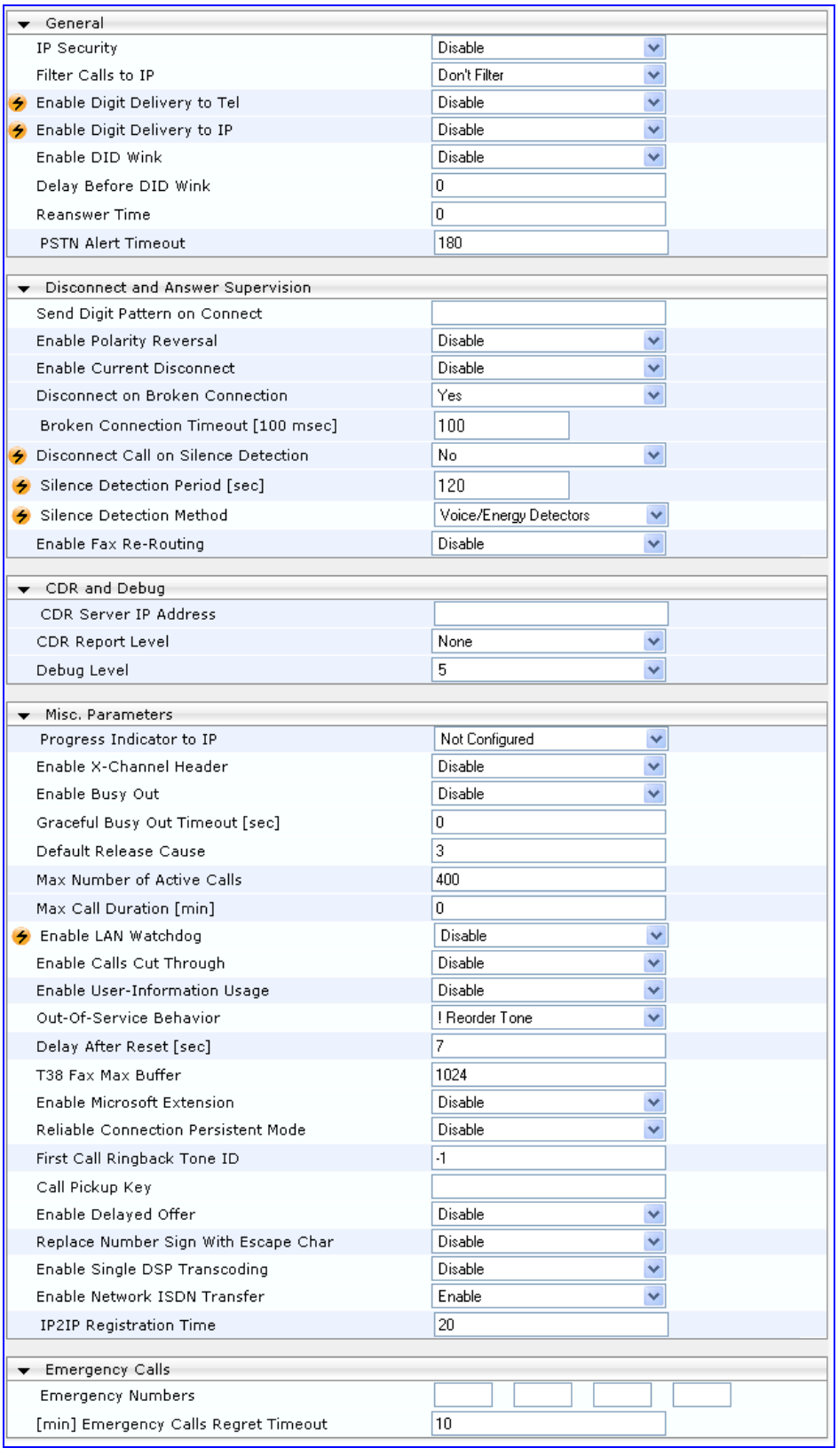
3.3.5.8.1 Configuring Advanced Parameters

The 'Advanced Parameters' page allows you to configure advanced SIP control parameters. For a description of the parameters appearing on this page, refer to "Configuration Parameters Reference" on page 247.

➢ **To configure the advanced general protocol parameters:**

1. Open the 'Advanced Parameters' page (**Configuration** tab > **Protocol Configuration** menu > **SIP Advanced Parameters** submenu > **Advanced Parameters** page item).

Figure 3-81: Advanced Parameters Page



2. Configure the parameters as required.
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

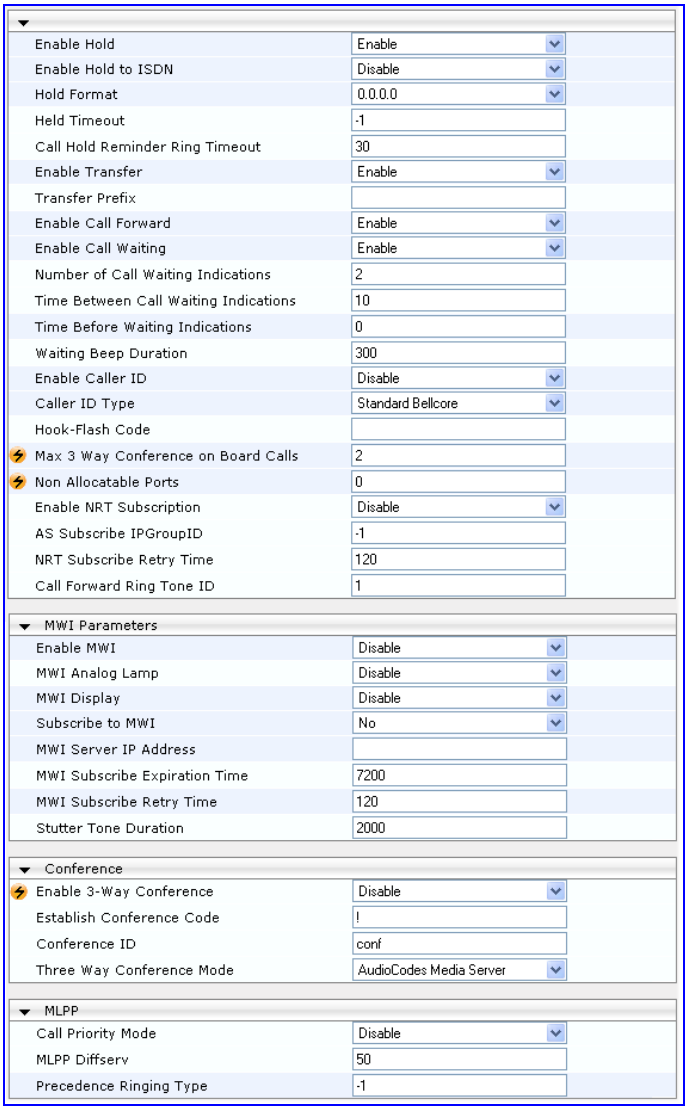
3.3.5.8.2 Configuring Supplementary Services

The 'Supplementary Services' page is used to configure parameters that are associated with supplementary services. For a description of the parameters appearing on this page, refer to "Configuration Parameters Reference" on page 247. For an overview on supplementary services, refer to "Working with Supplementary Services" on page 564.

➢ **To configure the supplementary services' parameters:**

1. Open the 'Supplementary Services' page (**Configuration** tab > **Protocol Configuration** menu > **SIP Advanced Parameters** submenu > **Supplementary Services** page item).

Figure 3-82: Supplementary Services Page



2. Configure the parameters as required.
3. Click the **Submit** button to save your changes, or click the **Subscribe to MWI** or **Unsubscribe to MWI** buttons to save your changes and to subscribe / unsubscribe to the MWI server.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

3.3.5.8.3 Configuring Metering Tones

The FXS interfaces can generate 12/16 KHz metering pulses towards the Tel side (e.g., for connection to a payphone or private meter). Tariff pulse rate is determined according to an internal table. This capability enables users to define different tariffs according to the source / destination numbers and the time-of-day. The tariff rate includes the time interval between the generated pulses and the number of pulses generated on answer.

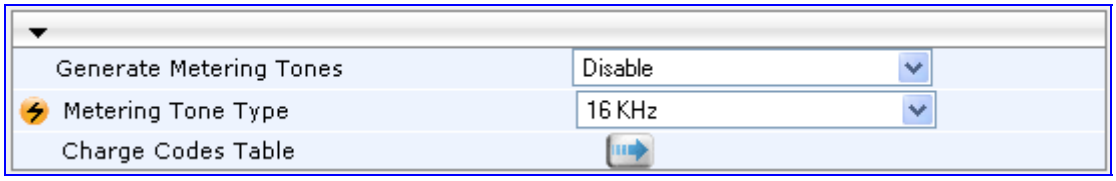
Note:

- The 'Metering Tones' page is available only for FXS interfaces.
- Charge Code rules can be assigned to routing rules in the 'Outbound IP Routing Table' (refer to "Configuring Outbound IP Routing Table" on page 158). When a new call is established, the 'Outbound IP Routing Table' is searched for the destination IP address. Once a route is located, the Charge Code (configured for that route) is used to associate the route with an entry in the 'Charge Codes' table.

➢ **To configure the Metering tones:**

1. Open the 'Metering Tones' page (**Configuration** tab > **Protocol Configuration** menu > **SIP Advanced Parameters** submenu > **Metering Tones** page item).

Figure 3-83: Metering Tones Page



2. Configure the Metering tones parameters as required. For a description of the parameters appearing on this page, refer to "Configuration Parameters Reference" on page 247.
3. Click the **Submit** button to save your changes.
4. To save the changes to the flash memory, refer to "Saving Configuration" on page 199.

If you set the 'Generate Metering Tones' parameter to 'Internal Table', access the 'Charge Codes Table' page by clicking the **Charge Codes Table** button. For a detailed description on configuring the Charge Codes table, refer to "Charge Codes Table" on page 139.

3.3.5.8.4 Configuring the Charge Codes Table

The 'Charge Codes Table' page is used to configure the metering tones (and their time interval) that the FXS interfaces generate to the Tel side. To associate a charge code to an outgoing Tel-to-IP call, use the 'Outbound IP Routing Table'.

Notes:

- The 'Charge Codes Table' page is available only for FXS interfaces.
- You can also configure the Charge Codes table using the *ini* file table parameter ChargeCode.

➢ **To configure the Charge Codes table:**

1. Access the 'Charge Codes Table' page (**Configuration** tab > **Protocol Configuration** menu > **SIP Advanced Parameters** submenu > **Charge Codes** page item). Alternatively, you can also access this page from the 'Metering Tones' page (refer to "Configuring Metering Tones" on page 139).

Figure 3-84: Charge Codes Table Page

Table Index: 0-4

Index	Time Period 1			Time Period 2			Time Period 3			Time Period 4		
	End Time	Pulse Interval	Pulses On Answer	End Time	Pulse Interval	Pulses On Answer	End Time	Pulse Interval	Pulses On Answer	End Time	Pulse Interval	Pulses On Answer
0	07	30	1	14	20	2	20	15	1	00	60	1
1	05	60	1	14	20	1	00	60	1			
2	00	60	1									
3												
4												

2. Define up to 25 different charge codes (each charge code is defined per row). Each charge code can include up to four different time periods in a day (24 hours). Each time period is composed of the following:
   - The end of the time period (in a 24 rounded-hour's format).
   - The time interval between pulses (in tenths of a second).
   - The number of pulses sent on answer.

   The first time period always starts at midnight (00). It is mandatory that the last time period of each rule ends at midnight (00). This prevents undefined time frames in a day. The device selects the time period by comparing the device 's current time to the end time of each time period of the selected Charge Code. The device generates the Number of Pulses on Answer once the call is connected and from that point on, it generates a pulse each Pulse Interval. If a call starts at a certain time period and crosses to the next, the information of the next time period is used.
3. Click the **Submit** button to save your changes.
4. To save the changes to the flash memory, refer to "Saving Configuration" on page 199.

3.3.5.8.5 Configuring Keypad Features

The 'Keypad Features' page enables you to activate and deactivate the following features directly from the connected telephone's keypad:

- Call Forward (refer to "Configuring Call Forward" on page 175)
- Caller ID Restriction (refer to "Configuring Caller Display Information" on page 173)
- Hotline (refer to "Configuring Automatic Dialing" on page 172)
- Call transfer
- Call waiting (refer to "Configuring Call Waiting" on page 177)
- Rejection of anonymous calls

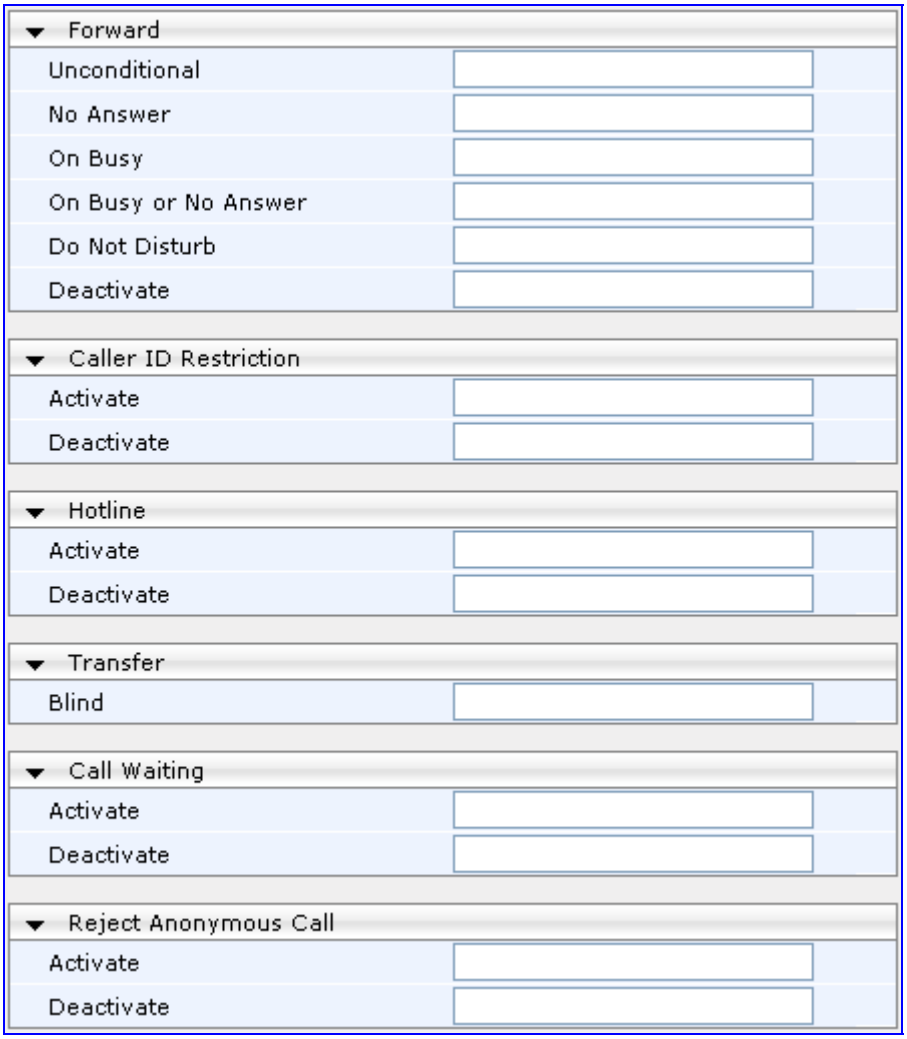
Notes:

- The 'Keypad Features' page is available only for FXS interfaces.
- The method used by the device to collect dialed numbers is identical to the method used during a regular call (i.e., max digits, interdigit timeout, digit map, etc.).
- The activation of each feature remains in effect until it is deactivated (i.e., not deactivated after a call).

➢ **To configure the keypad features**

1. Open the 'Keypad Features' page (**Configuration** tab > **Protocol Configuration** menu > **SIP Advanced Parameters** submenu > **Keypad Features** page item).

Figure 3-85: Keypad Features Page



2. Configure the keypad features as required. For a description of these parameters, refer to "Configuration Parameters Reference" on page 247.
3. Click the **Submit** button to save your changes.
4. To save the changes to the flash memory, refer to "Saving Configuration" on page 199.

3.3.5.9 Manipulation Tables

The **Manipulation Tables** submenu allows you to configure number manipulation and mapping of NPI/TON to SIP messages. This submenu includes the following items:

- General Settings (refer to "Configuring General Settings" on page 143)
- Manipulation tables (refer to "Configuring the Number Manipulation Tables" on page 143)**:**
  - Dest Number IP->Tel
  - Dest Number Tel->IP
  - Source Number IP->Tel
  - Source Number Tel->IP
- Redirect Number IP->Tel (refer to Configuring Redirect Number IP to Tel on page 149)
- Redirect Number Tel->IP (refer to "Configuring Redirect Number Tel to IP" on page 151)
- Phone Context (refer to "Mapping NPI/TON to SIP Phone-Context" on page 153)

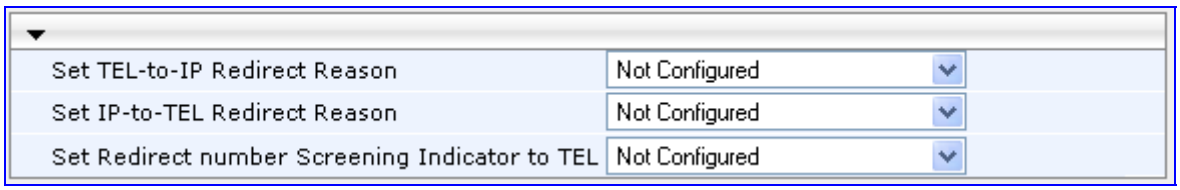
3.3.5.9.1 Configuring General Settings

The 'General Settings' page allows you to configure general manipulation parameters. For a description of the parameters appearing on this page, refer to "Configuration Parameters Reference" on page 247.

➢ **To configure the general manipulation parameters:**

1. Open the 'General Settings' page (**Configuration** tab > **Protocol Configuration** menu > **Manipulation Tables** submenu > **General Settings** page item).

Figure 3-86: General Settings Page



2. Configure the parameters as required.
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

3.3.5.9.2 Configuring the Number Manipulation Tables

The device provides four number manipulation tables for incoming (IP-to-Tel) and outgoing (Tel-to-IP) calls. These tables are used to modify the destination and source telephone numbers so that the calls can be routed correctly. For example, telephone number manipulation can be implemented for the following:

- Stripping or adding dialing plan digits from or to the number, respectively. For example, a user may need to first dial 9 before dialing the phone number to indicate an external line. This number 9 can then be removed by number manipulation before the call is setup.
- Allowing or blocking Caller ID information to be sent according to destination or source prefixes. For detailed information on Caller ID, refer to Configuring Caller Display Information on page 173.
- For digital modules only: Assigning Numbering Plan Indicator (NPI)/Type of Numbering (TON) to IP-to-Tel calls. The device can use a single global setting for NPI/TON classification or it can use the setting in the manipulation tables on a call-by-call basis.

The number manipulation is configured in the following tables:

- **For Tel-to-IP calls:**
  - Destination Phone Number Manipulation Table for Tel-to-IP Calls (NumberMapTel2IP *ini* file parameter) - up to 120 entries
  - Source Phone Number Manipulation Table for Tel-to-IP Calls (SourceNumberMapTel2IP *ini* file parameter) - up to 120 entries
- **For IP-to-Tel calls:**
  - Destination Phone Number Manipulation Table for IP-to-Tel Calls (NumberMapIP2Tel *ini* file parameter) - up to 100 entries
  - Source Phone Number Manipulation Table for IP-to-Tel Calls (SourceNumberMapIP2Tel *ini* file parameter) - up to 20 entries

The device matches manipulation rules starting at the top of the table. In other words, a rule at the top of the table takes precedence over a rule defined lower down in the table. Therefore, define more specific rules above more generic rules. For example, if you enter 551 in Index 1 and 55 in Index 2, the device applies rule 1 to numbers that start with 551 and applies rule 2 to numbers that start with 550, 552, 553, and so on untill 559. However, if you enter 55 in Index 1 and 551 in Index 2, the device applies rule 1 to all numbers that start with 55, including numbers that start with 551.

Notes:

- Number manipulation can occur before or after a routing decision is made. For example, you can route a call to a specific Trunk Group according to its original number, and then you can remove or add a prefix to that number before it is routed. To determine when number manipulation is performed, configure the 'IP to Tel Routing Mode' parameter (RouteModeIP2Tel) described in "Configuring the Inbound IP Routing Table" on page 163, and 'Tel to IP Routing Mode' parameter (RouteModeTel2IP) described in "Configuring the Outbound IP Routing Table" on page 158.
- The manipulation rules are executed in the following order:
  1. Stripped digits from left.
  2. Stripped digits from right.
  3. Number of digits to leave.
  4. Prefix to add.
  5. Suffix to add.
- The manipulation rules can be applied to any incoming call whose source IP address (if applicable), source Trunk Group (if applicable), source IP Group (if applicable), destination number prefix and source number prefix matches the values defined in the 'Source IP Address', 'Source Trunk Group', 'Source IP Group', 'Destination Prefix', and 'Source Prefix' fields respectively. The number manipulation can be performed using a combination of each of the above criteria, or using each criterion independently.
- For available notations representing multiple numbers/digits for destination and source prefixes, refer to "Dialing Plan Notation for Routing and Manipulation" on page 485.
- For configuring number manipulation using *ini* file table parameters NumberMapIP2Tel, NumberMapTel2IP, SourceNumberMapIP2Tel, and SourceNumberMapTel2IP, refer to "Number Manipulation and Routing Parameters" on page 428.

➢ **To configure the Number Manipulation tables:**

1. Open the required 'Number Manipulation' page (**Configuration** tab > **Protocol Configuration** menu > **Manipulation Tables** submenu > **Dest Number IP->Tel**, **Dest Number Tel->IP**, **Source Number IP->Tel**, or **Source Number Tel->IP** page item); the relevant Manipulation table page is displayed (e.g., 'Source Phone Number Manipulation Table for Tel→IP Calls' page).

Figure 3-87: Source Phone Number Manipulation Table for Tel-to-IP Calls

Index	Source Trunk Group	Source IP Group	Destination Prefix	Source Prefix	Stripped Digits From Left
1	-1	2	03	201	0
2	0	0		1001	4
3	-1	-1	*	123451001#	0
4	-1	-1	*	[30-40]x	0
5	-1	-1	[6,7,8]	2001	5

Stripped Digits From Right	Prefix to Add	Suffix to Add	Number of Digits to Leave	Presentation
0	971		255	Allowed
0	5	23	255	Restricted
0		8	4	Not Configured
1	2		255	Not Configured
0	3		255	Not Configured

The figure above shows an example of the use of manipulation rules for Tel-to-IP source phone number manipulation:

- **Index 1:** When the destination number has the prefix 03 (e.g., 035000), source number prefix 201 (e.g., 20155), and from source IP Group ID 2, the source number is changed to, for example, 97120155.
- **Index 2:** When the source number has prefix 1001 (e.g., 1001876), it is changed to 587623.
- **Index 3:** When the source number has prefix 123451001 (e.g., 1234510012001), it is changed to 20018.
- **Index 4:** When the source number has prefix from 30 to 40 and a digit (e.g., 3122), it is changed to 2312.
- **Index 5:** When the destination number has the prefix 6, 7, or 8 (e.g., 85262146), source number prefix 2001, it is changed to 3146.

2. From the 'Table Index' drop-down list, select the range of entries that you want to edit.
3. Configure the Number Manipulation table according to the table below.
4. Click the **Submit** button to save your changes.
5. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

Table 3-24: Number Manipulation Parameters Description

Parameter	Description
Source Trunk Group **[_SrcTrunkGroupID]**	The source Trunk Group ID for Tel-to-IP calls. To denote any Trunk Group, leave this field empty. **Notes:** ▪ The value -1 indicates that it is ignored in the rule. ▪ This parameter is available only in the 'Source Phone Number

Parameter	Description
	Manipulation Table for Tel -> IP Calls' and 'Destination Phone Number Manipulation Table for Tel -> IP Calls' pages. ▪ For IP-to-IP call routing, this parameter is not required (i.e., leave the field empty).
Source IP Group **[_SrcIPGroupID]**	The IP Group from where the IP-to-IP call originated. Typically, this IP Group of an incoming INVITE is determined/classified using the 'Inbound IP Routing Table'. If not used (i.e., any IP Group), simply leave the field empty. **Notes:** ▪ The value -1 indicates that it is ignored in the rule. ▪ This parameter is available only in the 'Source Phone Number Manipulation Table for Tel -> IP Calls' and 'Destination Phone Number Manipulation Table for Tel -> IP Calls' pages. ▪ If this Source IP Group has a Serving IP Group, then all calls originating from this Source IP Group is sent to the Serving IP Group. In this scenario, this table is used only if the parameter PreferRouteTable is set to 1.
Web: Destination Prefix EMS: Prefix **[_DestinationPrefix]**	Destination (called) telephone number prefix. An asterisk (*) represents any number.
Web/EMS: Source Prefix **[_SourcePrefix]**	Source (calling) telephone number prefix. An asterisk (*) represents any number.
Web/EMS: Source IP **[_SourceAddress]**	Source IP address of the caller (obtained from the Contact header in the INVITE message). **Notes:** ▪ This parameter is applicable only to the Number Manipulation tables for IP-to-Tel calls. ▪ The source IP address can include the 'x' wildcard to represent single digits. For example: 10.8.8.xx represents all IP addresses between 10.8.8.10 to 10.8.8.99. ▪ The source IP address can include the asterisk (*) wildcard to represent any number between 0 and 255. For example, 10.8.8.* represents all IP addresses between 10.8.8.0 and 10.8.8.255.
Web: Stripped Digits From Left EMS: Number Of Stripped Digits **[_RemoveFromLeft]**	Number of digits to remove from the left of the telephone number prefix. For example, if you enter 3 and the phone number is 5551234, the new phone number is 1234.
Web: Stripped Digits From Right EMS: Number Of Stripped Digits **[_RemoveFromRight]**	Number of digits to remove from the right of the telephone number prefix. For example, if you enter 3 and the phone number is 5551234, the new phone number is 5551.
Web: Prefix to Add EMS: Prefix/Suffix To Add **[_Prefix2Add]**	The number or string that you want added to the front of the telephone number. For example, if you enter '9' and the phone number is 1234, the new number is 91234.

Parameter	Description
Web: Suffix to Add EMS: Prefix/Suffix To Add **[_Suffix2Add]**	The number or string that you want added to the end of the telephone number. For example, if you enter '00' and the phone number is 1234, the new number is 123400.
Web/EMS: Number of Digits to Leave **[_LeaveFromRight]**	The number of digits that you want to retain from the right of the phone number.
Web: NPI EMS: Number Plan **[_NumberPlan]**	The Numbering Plan Indicator (NPI) assigned to this entry. ▪ **[0]** Unknown (default) ▪ **[9]** Private ▪ **[1]** E.164 Public ▪ **[-1]** Not Configured = value received from PSTN/IP is used **Notes:** ▪ This parameter is applicable only to Number Manipulation tables for IP-to-Tel calls. ▪ For a detailed list of the available NPI/TON values, refer to Numbering Plans and Type of Number on page 155
Web: TON EMS: Number Type **[_NumberType]**	The Type of Number (TON) assigned to this entry. ▪ If you selected 'Unknown' for the NPI, you can select Unknown **[0].** ▪ If you selected 'Private' for the NPI, you can select Unknown **[0],** Level 2 Regional **[1],** Level 1 Regional **[2],** PISN Specific **[3]** or Level 0 Regional (Local) **[4].** ▪ If you selected 'E.164 Public' for the NPI, you can select Unknown **[0],** International **[1],** National **[2],** Network Specific **[3],** Subscriber **[4]** or Abbreviated **[6].** **Notes:** ▪ This parameter is applicable only to Number Manipulation tables for IP-to-Tel calls. ▪ The default is 'Unknown'.
Web: Presentation EMS: Is Presentation Restricted **[_IsPresentationRestricted]**	Determines whether Caller ID is permitted: ▪ Not Configured = privacy is determined according to the Caller ID table (refer to "Configuring Caller Display Information" on page 173). ▪ Allowed = sends Caller ID information when a call is made using these destination / source prefixes. ▪ Restricted = restricts Caller ID information for these prefixes. **Notes:** ▪ Only applicable to Number Manipulation tables for source number manipulation. ▪ If 'Presentation' is set to 'Restricted' and 'Asserted Identity Mode' is set to 'P-Asserted', the From header in the INVITE message includes the following: From: 'anonymous' <sip: anonymous@anonymous.invalid> and 'privacy: id' header.

3.3.5.9.3 Configuring Redirect Number IP to Tel

The 'Redirect Number IP > Tel' page allows you to configure IP-to-Tel redirect number manipulation rules. This feature allows you to manipulate the value of the received SIP Diversion, Resource-Priority, or History-Info header, which is then added to the Redirecting Number Information Element (IE) in the ISDN Setup message, sent to the Tel side.

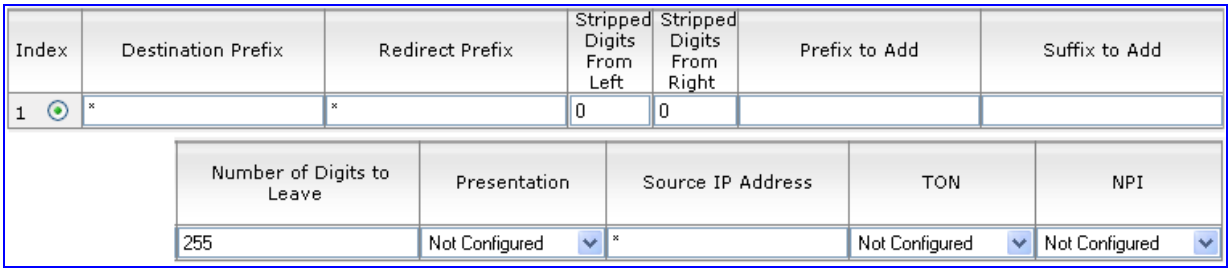
Notes:

- You can also configure the Redirect Number IP to Tel table using the *ini* file parameter RedirectNumberMapIp2Tel (refer to "Number Manipulation and Routing Parameters" on page 428).
- If the characteristics DestinationPrefix, RedirectPrefix, and/or SourceAddress match the incoming SIP message, manipulation is performed according to the configured manipulation rule.
- The manipulation rules are executed in the following order: RemoveFromLeft, RemoveFromRight, LeaveFromRight, Prefix2Add, and then Suffix2Add.
- The DestinationNumber and RedirectPrefix parameters are used before any manipulation has been performed on them.
- Redirect manipulation is performed only after the parameter CopyDest2RedirectNumber.

➢ **To configure the Redirect Number IP-to-Tel manipulation rules:**

1. Open the 'Redirect Number IP > Tel' page (**Configuration** tab > **Protocol Configuration** menu > **Manipulation Tables** submenu > **Redirect Number IP > Tel** page item).

Figure 3-88: Reditect Number IP to Tel Page



2. Configure the rules according to the table below.
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

Table 3-25: Redirect Number IP to Tel Parameters Description

Parameter	Description
Web/EMS: Destination Prefix **[_DestinationPrefix]**	Destination (called) telephone number prefix. An asterisk (*) represents any number.
Web/EMS: Redirect Prefix **[_RedirectPrefix]**	Redirect telephone number prefix. An asterisk (*) represents any number.

Parameter	Description
Web: Stripped Digits From Left EMS: Remove From Left **[_RemoveFromLeft]**	Number of digits to remove from the left of the telephone number prefix. For example, if you enter 3 and the phone number is 5551234, the new phone number is 1234.
Web: Stripped Digits From Right EMS: Remove From Right **[_RemoveFromRight]**	Number of digits to remove from the right of the telephone number prefix. For example, if you enter 3 and the phone number is 5551234, the new phone number is 5551.
Web/EMS: Prefix to Add **[_Prefix2Add]**	The number or string that you want added to the front of the telephone number. For example, if you enter '9' and the phone number is 1234, the new number is 91234.
Web/EMS: Suffix to Add **[_Suffix2Add]**	The number or string that you want added to the end of the telephone number. For example, if you enter '00' and the phone number is 1234, the new number is 123400.
Web/EMS: Number of Digits to Leave **[_LeaveFromRight]**	The number of digits that you want to retain from the right of the phone number.
Web: Presentation EMS: Is Presentation Restricted **[_IsPresentationRestricted]**	Determines whether Caller ID is permitted: ▪ Not Configured = privacy is determined according to the Caller ID table (refer to "Configuring Caller Display Information" on page 173). ▪ Allowed = sends Caller ID information when a call is made using these destination / source prefixes. ▪ Restricted = restricts Caller ID information for these prefixes. **Notes:** ▪ If 'Presentation' is set to 'Restricted' and 'Asserted Identity Mode' is set to 'P-Asserted', the From header in the INVITE message includes the following: From: 'anonymous' <sip: anonymous@anonymous.invalid> and 'privacy: id' header.
Web/EMS: Source IP Address **[_SourceAddress]**	Source IP address of the caller (obtained from the Contact header in the INVITE message). **Note:** The source IP address can include the following wildcards: ✓ "x": represents single digits. For example, 10.8.8.xx depicts all addresses between 10.8.8.10 and 10.8.8.99. ✓ "*": represents any number between 0 and 255. For example, 10.8.8.* depicts all addresses between 10.8.8.0 and 10.8.8.255.
Web: TON EMS: Number Type **[_NumberType]**	The Type of Number (TON) assigned to this entry. The default is 'Unknown' **[0]**. ▪ If you select 'Unknown' for the NPI, you can select Unknown **[0]**. ▪ If you select 'Private' for the NPI, you can select Unknown **[0]**, International **[1]**, National **[2]**, Network Specific **[3]** or Subscriber **[4]**. ▪ If you select 'E.164 Public' for the NPI, you can select Unknown **[0]**, International **[1]**, National **[2]**, Network Specific **[3]**, Subscriber **[4]** or Abbreviated **[6]**.

Parameter	Description
Web: NPI EMS: Number Plan **[_NumberPlan]**	The Numbering Plan Indicator (NPI) assigned to this entry. ▪ **[0]** Unknown (default) ▪ **[9]** Private ▪ **[1]** E.164 Public ▪ **[-1]** Not Configured = value received from PSTN/IP is used **Note:** For a detailed list of the available NPI/TON values, refer to "Numbering Plans and Type of Number" on page 155

3.3.5.9.4 Configuring Redirect Number Tel to IP

The 'Redirect Number Tel > IP' page allow you to configure Tel-to-IP Redirect Number manipulation rules. This feature manipulates the prefix of the redirect number received from the PSTN for the outgoing SIP Diversion, Resource-Priority, or History-Info header that is sent to IP.

Notes:

- You can also configure the Redirect Number Tel to IP table using the *ini* file parameter RedirectNumberMapTel2Ip (refer to "Number Manipulation and Routing Parameters" on page 428).
- If the characteristics DestinationPrefix, RedirectPrefix, and/or SourceAddress match the incoming SIP message, manipulation is performed according to the configured manipulation rule.
- The manipulation rules are executed in the following order: RemoveFromLeft, RemoveFromRight, LeaveFromRight, Prefix2Add, and then Suffix2Add.
- The DestinationNumber and RedirectPrefix parameters are used before any manipulation has been performed on them.
- Redirect manipulation is performed only after the parameter CopyDest2RedirectNumber.

➢ **To configure the redirect Tel to IP table:**

1. Open the 'Redirect Number Tel > IP' page (**Configuration** tab > **Protocol Configuration** menu > **Manipulation Tables** submenu > **Redirect Number Tel > IP** page item).

Figure 3-89: Redirect Number Tel to IP Page

Index	Source Trunk Group	Source IP Group	Destination Prefix	Redirect Prefix	Stripped Digits From Left	Stripped Digits From Right	Prefix to Add
1	-1	-1	*	555	3	0	9

Suffix to Add	Number of Digits to Leave	Presentation
	255	Not Configured

The figure below shows an example configuration in which the redirect prefix "555" is manipulated. According to the configured rule, if for example the number 5551234 is received, after manipulation the device sends the number to IP as 91234.

2. Configure the redirect number Tel to IP rules according to the table below.
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

Table 3-26: Redirect Number Tel to IP Parameters Description

Parameter	Description
Source Trunk Group **[_SrcTrunkGroupID]**	The Trunk Group from where the Tel call is received. To denote any Trunk Group, leave this field empty. **Notes:** ▪ The value -1 indicates that it is ignored in the rule. ▪ For IP-to-IP call routing, this parameter is not required (i.e., leave the field empty).
Source IP Group **[_SrcIPGroupID]**	The IP Group from where the IP-to-IP call originated. Typically, the IP Group of an incoming INVITE is determined/classified using the 'Inbound IP Routing Table'. If not used (i.e., any IP Group), simply leave the field empty. **Notes:** ▪ The value -1 indicates that it is ignored in the rule. ▪ This parameter is applicable only to the IP-to-IP application.
Web/EMS: Destination Prefix **[_DestinationPrefix]**	Destination (called) telephone number prefix. An asterisk (*) represents any number.
Web/EMS: Redirect Prefix **[_RedirectPrefix]**	Redirect telephone number prefix. An asterisk (*) represents any number.
Web: Stripped Digits From Left EMS: Remove From Left **[_RemoveFromLeft]**	Number of digits to remove from the left of the telephone number prefix. For example, if you enter 3 and the phone number is 5551234, the new phone number is 1234.
Web: Stripped Digits From Right EMS: Remove From Right **[_RemoveFromRight]**	Number of digits to remove from the right of the telephone number prefix. For example, if you enter 3 and the phone number is 5551234, the new phone number is 5551.
Web/EMS: Prefix to Add **[_Prefix2Add]**	The number or string that you want added to the front of the telephone number. For example, if you enter '9' and the phone number is 1234, the new number is 91234.
Web/EMS: Suffix to Add **[_Suffix2Add]**	The number or string that you want added to the end of the telephone number. For example, if you enter '00' and the phone number is 1234, the new number is 123400.
Web/EMS: Number of Digits to Leave **[_LeaveFromRight]**	The number of digits that you want to retain from the right of the phone number.
Web: Presentation EMS: Is Presentation Restricted **[_IsPresentationRestricted]**	Determines whether Caller ID is permitted: ▪ Not Configured = privacy is determined according to the Caller ID table (refer to "Configuring Caller Display Information" on page 173).

Parameter	Description
	▪ Allowed = sends Caller ID information when a call is made using these destination / source prefixes. ▪ Restricted = restricts Caller ID information for these prefixes. **Notes:** ▪ If 'Presentation' is set to 'Restricted' and 'Asserted Identity Mode' is set to 'P-Asserted', the From header in the INVITE message includes the following: From: 'anonymous' <sip: anonymous@anonymous.invalid> and 'privacy: id' header.

3.3.5.9.5 Mapping NPI/TON to SIP Phone-Context

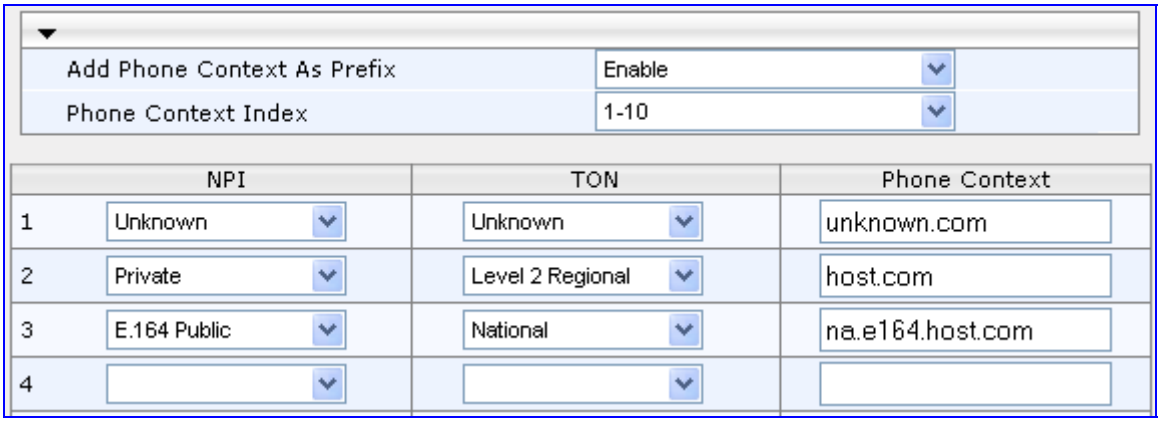
The 'Phone-Context Table' page is used to map Numbering Plan Indication (NPI) and Type of Number (TON) to the SIP Phone-Context parameter. When a call is received from the ISDN/Tel, the NPI and TON are compared against the table and the matching Phone-Context value is used in the outgoing SIP INVITE message. The same mapping occurs when an INVITE with a Phone-Context attribute is received. The Phone-Context parameter appears in the standard SIP headers where a phone number is used (Request-URI, To, From, Diversion).

For example, for a Tel-to-IP call with NPI/TON set as E164 National (values 1/2), the device sends the outgoing SIP INVITE URI with the following settings: “sip:12365432;phone-context= na.e.164.nt.com“. This is configured for entry 3 in the figure below. In the opposite direction (IP-to-Tel call), if the incoming INVITE contains this Phone-Context (e.g. "phone-context= na.e.164.nt.com"), the NPI/TON of the called number in the outgoing SETUP message is changed to E164 National.

➢ **To configure the Phone-Context tables:**

1. Open the 'Phone Context Table' page (**Configuration** tab > **Protocol Configuration** menu > **Manipulation Tables** submenu > **Phone Context** page item).

Figure 3-90: Phone Context Table Page



2. Configure the Phone Context table according to the table below.
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

Notes:

- Several rows with the same NPI-TON or Phone-Context are allowed. In such a scenario, a Tel-to-IP call uses the first match.
- Phone-Context '+' is a unique case as it doesn't appear in the Request-URI as a Phone-Context parameter. Instead, it's added as a prefix to the phone number. The '+' isn't removed from the phone number in the IP-to-Tel direction.
- You can also configure the Phone Context table using the *ini* file table parameter PhoneContext (refer to "Number Manipulation and Routing Parameters" on page 428).

Table 3-27: Phone-Context Parameters Description

Parameter	Description
Add Phone Context As Prefix **[AddPhoneContextAsPrefix]**	Determines whether the received Phone-Context parameter is added as a prefix to the outgoing ISDN SETUP message (digital interfaces) with Called and Calling numbers. ▪ **[0]** Disable = Disable (default). ▪ **[1]** Enable = Enable.
NPI	Select the Number Plan assigned to this entry. ▪ **[0]** Unknown = Unknown (default) ▪ **[1]** E.164 Public = E.164 Public ▪ **[9]** Private = Private For a detailed list of the available NPI/TON values, refer to Numbering Plans and Type of Number on page 155.
TON	Select the Type of Number assigned to this entry. ▪ If you selected Unknown as the NPI, you can select Unknown **[0]**. ▪ If you selected Private as the NPI, you can select one of the following: ✓ **[0]** Unknown ✓ **[1]** Level 2 Regional ✓ **[2]** Level 1 Regional ✓ **[3]** PSTN Specific ✓ **[4]** Level 0 Regional (Local) ▪ If you selected E.164 Public as the NPI, you can select one of the following: ✓ **[0]** Unknown ✓ **[1]** International ✓ **[2]** National ✓ **[3]** Network Specific ✓ **[4]** Subscriber ✓ **[6]** Abbreviated
Phone Context	The Phone-Context SIP URI parameter.

3.3.5.9.6 Numbering Plans and Type of Number

The IP-to-Tel destination or source number manipulation tables allow you to classify numbers by their Numbering Plan Indication (NPI) and Type of Number (TON). The device supports all NPI/TON classifications used in the standard. The list of ISDN ETSI NPI/TON values is shown in the following table:

Table 3-28: NPI/TON Values for ISDN ETSI

NPI	TON	Description
Unknown **[0]**	Unknown **[0]**	A valid classification, but one that has no information about the numbering plan.
E.164 Public **[1]**	Unknown **[0]**	A public number in E.164 format, but no information on what kind of E.164 number.
	International **[1]**	A public number in complete international E.164 format, e.g., 16135551234.
	National **[2]**	A public number in complete national E.164 format, e.g., 6135551234.
	Subscriber **[4]**	A public number in complete E.164 format representing a local subscriber, e.g., 5551234.
Private **[9]**	Unknown **[0]**	A private number, but with no further information about the numbering plan.
	Level 2 Regional **[1]**	
	Level 1 Regional **[2]**	A private number with a location, e.g., 3932200.
	PISN Specific **[3]**	
	Level 0 Regional (local) **[4]**	A private local extension number, e.g., 2200.

For NI-2 and DMS-100 ISDN variants, the valid combinations of TON and NPI for calling and called numbers include (Plan/Type):

- 0/0 - Unknown/Unknown
- 1/1 - International number in ISDN/Telephony numbering plan
- 1/2 - National number in ISDN/Telephony numbering plan
- 1/4 - Subscriber (local) number in ISDN/Telephony numbering plan
- 9/4 - Subscriber (local) number in Private numbering plan

3.3.5.10 Routing Tables

The **Routing Tables** submenu allows you to configure call routing rules. This submenu includes the following page items:

- Alternative Routing (refer to "Configuring Reasons for Alternative Routing" on page 156)
- Routing General Parameters (refer to "Configuring Routing General Parameters" on page 157)
- Tel to IP Routing (refer to "Configuring the Outbound IP Routing Table" on page 158)
- IP to Trunk Group Routing (refer to "Configuring the Inbound IP Routing Table" on page 163)
- Internal DNS Table (refer to "Configuring the Internal DNS Table" on page 167)
- Internal SRV Table (refer to "Configuring the Internal SRV Table" on page 168)
- Release Cause Mapping (refer to Configuring Release Cause Mapping on page 169)
- Forward on Busy Trunk Dest (refer to "Configuring Call Forward upon Busy Trunk" on page 169)

3.3.5.10.1 Configuring Reasons for Alternative Routing

The 'Reasons for Alternative Routing' page allows you to define up to four different call release (termination) reasons for IP-to-Tel call releases and for Tel-to-IP call releases. If a call is released as a result of one of these reasons, the device tries to find an alternative route for that call. The device supports up to two different alternative routes.

The release reasons depends on the call direction:

- **Release reason for IP-to-Tel calls:** provided in Q.931 notation. As a result of a release reason, an alternative Trunk Group is provided. For defining an alternative Trunk Group, refer to "Configuring the Inbound IP Routing Table" on page 163.

  This call release reason type can be configured, for example, when the destination is busy and release reason #17 is issued or for other call releases that issue the default release reason (#3) - refer to the parameter DefaultReleaseCause.

- **Release reason for Tel-to-IP calls:** provided in SIP 4xx, 5xx, and 6xx response codes. As a result of a release reason, an alternative IP address is provided. For defining an alternative IP address, refer to "Configuring the Outbound IP Routing Table" on page 158.

  This call release reason type can be configured, for example, when there is no response to an INVITE message (after INVITE re-transmissions), the device issues an internal 408 'No Response' implicit release reason.

The device also plays a tone to the endpoint whenever an alternative route is used. This tone is played for a user-defined time, configured by the *ini* file parameter AltRoutingToneDuration.

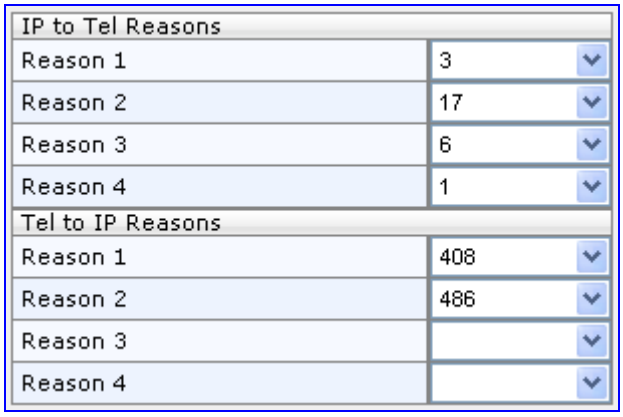
Notes:

- To enable alternative routing using the IP-to-Tel routing table, configure the parameter RedundantRoutingMode to 1 (default).
- The reasons for alternative routing for Tel-to-IP calls also apply for Proxies (if the parameter RedundantRoutingMode is set to 2).
- You can also configure alternative routing using the *ini* file table parameters AltRouteCauseTel2IP and AltRouteCauseIP2Tel (refer to "Number Manipulation and Routing Parameters" on page 428).

➢ **To configure the reasons for alternative routing:**

1. Open the 'Reasons for Alternative Routing' page (**Configuration** tab > **Protocol Configuration** menu > **Routing Tables** submenu > **Alternative Routing** page item).

Figure 3-91: Reasons for Alternative Routing Page



2. In the 'IP to Tel Reasons' group, select up to four different call failure reasons that invoke an alternative IP-to-Tel routing.
3. In the 'Tel to IP Reasons' group, select up to four different call failure reasons that invoke an alternative Tel-to-IP routing.
4. Click the **Submit** button to save your changes.
5. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

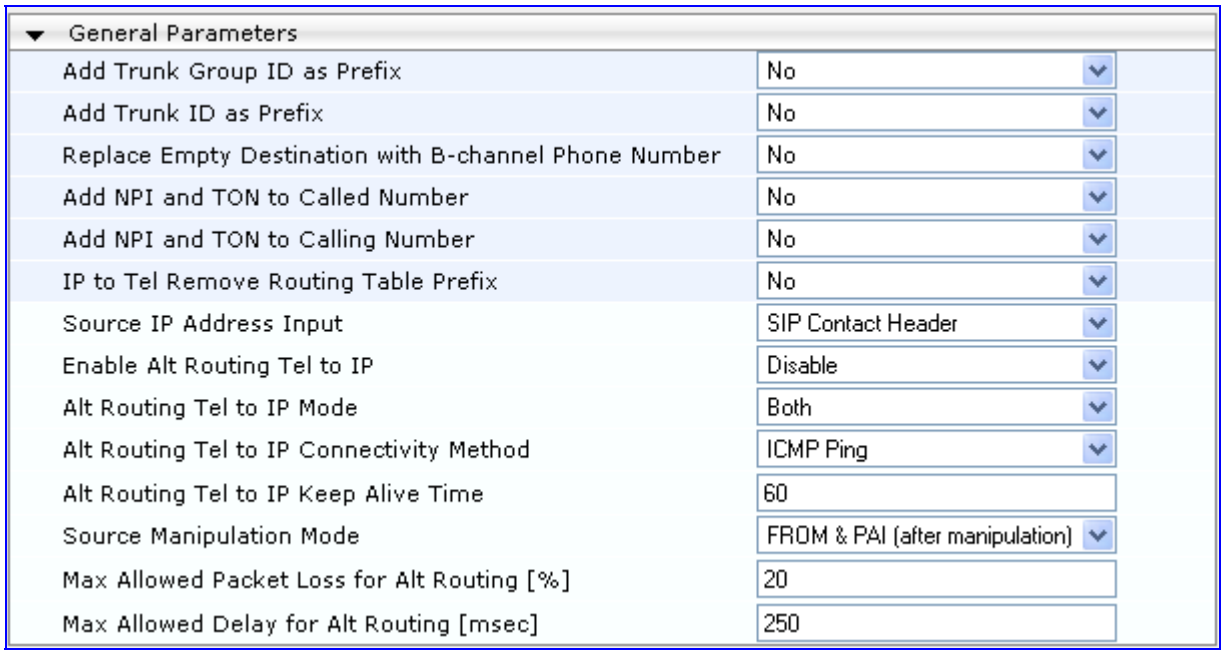
3.3.5.10.2 Configuring General Routing Parameters

The 'Routing General Parameters' page allows you to configure the general routing parameters. For a description of these parameters, refer to "Configuration Parameters Reference" on page 247.

➢ **To configure the general routing parameters:**

1. Open the 'Routing General Parameters' page (**Configuration** tab > **Protocol Configuration** menu > **Routing Tables** submenu > **Routing General Parameters** page item).

Figure 3-92: Routing General Parameters Page



2. Configure the parameters as required.
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

3.3.5.10.3 Configuring the Outbound IP Routing Table

The 'Outbound IP Routing Table' page provides a table for configuring up to 180 outbound IP call routing rules. The device uses these rules to route calls (Tel or IP) to IP destinations (when a proxy server is not used for routing).

This table provides two main areas for defining a routing rule:

- **Matching Characteristics:** user-defined characteristics of the incoming call are defined in this area. If the characteristics match a table entry, the rule is used to route the call. One or more characteristics can be defined for the rule such as source IP Group (to which the call belongs), Trunk Group (from where the call is received), source (calling)/destination (called) telephone number prefix, source/destination Request URI host name prefix.
- **Destination:** user-defined IP destination. If the call matches the characteristics, the device routes the call to this destination. The destination can be defined as an IP address (or Fully Qualified Domain Name/FQDN) or IP Group. If defined as a specific

IP Group, the call is routed to the Proxy Set (IP address) associated with the IP Group. If the number dialed does not match these characteristics, the call is not made.

When using a proxy server, you don't need to configure this table unless you require one of the following:

- Fallback routing if communication is lost with proxy servers.
- IP Security feature (enabled using the SecureCallFromIP parameter): the device accepts only received calls whose source IP address is defined in this routing table.
- Filter Calls to IP feature: the device checks this routing table before a call is routed to the proxy. However, if the number is not allowed, i.e., the number does not exist in the table or a Call Restriction (see below) routing rule is applied, the call is released.
- Obtain different SIP URI host names (per called number).
- Assign IP Profiles.

Note that for this table to take precedence over a proxy for routing calls, you need to set the parameter PreferRouteTable to 1. The device checks the 'Destination IP Address' field in this table for a match with the outgoing call. A proxy is used only if a match is not found.

Possible uses for configuring routing rules in this table (in addition to those listed above when using a proxy), include the following:

- Call Restriction: rejects all outgoing calls whose routing rule is associated with the destination IP address 0.0.0.0.
- Always Use Routing Table feature (enabled using the AlwaysUseRouteTable parameter): even if a proxy server is used, the SIP URI host name in the sent INVITE message is obtained from this table. Using this feature, you can assign a different SIP URI host name for different called and/or calling numbers.
- Assign IP Profiles to destination addresses (also when a proxy is used).
- Alternative Routing (when a proxy isn't used): an alternative IP destination can be configured for a specific call type. To associate an alternative IP address to a called telephone number prefix, assign it with an additional entry (with a different IP address), or use an FQDN that resolves into two IP addresses. The call is sent to the alternative destination when one of the following occurs:
  - Ping to the initial destination is unavailable, poor QoS (delay or packet loss, calculated according to previous calls) is detected or a DNS host name is unresolved. For detailed information on Alternative Routing, refer to "Configuring Alternative Routing (Based on Connectivity and QoS" on page 554).
  - A release reason defined in the 'Reasons for Alternative Tel to IP Routing' table is received (refer to "Configuring Reasons for Alternative Routing" on page 156).

  Alternative routing is commonly implemented when there is no response to an INVITE message (after INVITE retransmissions). The device then issues an internal 408 'No Response' implicit release reason. If this reason is included in the 'Reasons for Alternative Routing' table, the device immediately initiates a call to the alternative destination using the next matched entry in this routing table. Note that if a domain name in this table is resolved into two IP addresses, the timeout for INVITE retransmissions can be reduced by using the parameter 'Number of RTX Before Hotswap'.

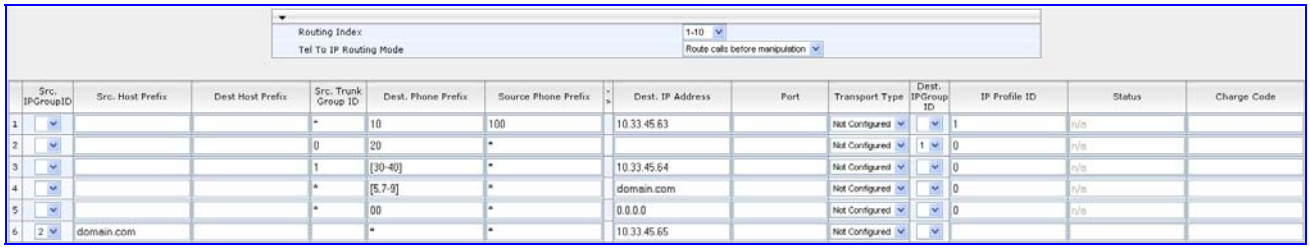
Notes:

- If the alternative routing destination is the device itself, the call can be configured to be routed to the PSTN. This feature is referred to as 'PSTN Fallback'. For example, if poor voice quality occurs over the IP network, the call is rerouted through the legacy telephony system (PSTN).
- Outbound IP routing can be performed before or after number manipulation rules are applied. This is configured using the RouteModeTel2IP parameter, as described below.
- You can also configure this table using the *ini* file table parameter Prefix (refer to "Number Manipulation and Routing Parameters" on page 428).

➢ **To configure outbound IP routing rules:**

1. Open the 'Outbound IP Routing Table' page (**Configuration** tab > **Protocol Configuration** menu > **Routing Tables** submenu > **Tel to IP Routing** page item).

Figure 3-93: Outbound IP Routing Table Page


Routing Index
1-10
Tel To IP Routing Mode
Route calls before manipulation
Src. IPGroup ID
Src. Host Prefix
Dest Host Prefix
Src. Trunk Group ID
Dest. Phone Prefix
Source Phone Prefix
Dest. IP Address
Port
Transport Type
Dest. IPGroup ID
IP Profile ID
Status
Charge Code
1 10 100 10.33.45.63 Not Configured 1
2 0 20 * Not Configured 1 0
3 1 [30-40] * 10.33.45.64 Not Configured 0
4 * [5,7-9] * domain.com Not Configured 0
5 * 00 * 0.0.0.0 Not Configured 0
6 2 domain.com * * 10.33.45.65 Not Configured

The figure above shows the following configured outbound IP routing rules:

- **Rule 1:** If the called phone prefix is 10 and the caller's phone prefix is 100, the call is assigned settings configured for IP Profile ID 1 and sent to IP address 10.33.45.63.
- **Rule 2:** If the called phone prefix is 20 and the caller is all prefixes (*), the call is sent to the destination according to IP Group 1 (which in turn is associated with a Proxy Set ID providing the IP address).
- **Rule 3:** If the called phone prefix is between 30 and 40, and the caller belongs to Trunk Group ID 1, the call is sent to IP address 10.33.45.64.
- **Rule 4:** If the called phone prefix is either 5, 7, 8, or 9 and the caller is all (*), the call is sent to domain.com.
- **Rule 5:** If the called phone prefix is 00 and the caller is all (*), the call is discarded.
- Rule 6: If an incoming IP call pertaining to Source IP Group 2 with domain.com as source host prefix in its Request URI, the IP call is sent to IP address 10.33.45.65.

2. From the 'Routing Index' drop-down list, select the range of entries that you want to add.
3. Configure the outbound IP routing rules according to the table below.
4. Click the **Submit** button to apply your changes.
5. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

Table 3-29: Outbound IP Routing Table Parameters

Parameter	Description
Web/EMS: Tel to IP Routing Mode **[RouteModeTel2IP]**	Determines whether to route received calls to an IP destination before or after manipulation of the destination number. ▪ **[0]** Route calls before manipulation = Calls are routed before the number manipulation rules are applied (default). ▪ **[1]** Route calls after manipulation = Calls are routed after the number manipulation rules are applied. **Notes:** ▪ This parameter is not applicable if outbound proxy routing is used. ▪ For number manipulation, refer to "Configuring the Number Manipulation Tables" on page 143.
Web: Src. IPGroupID EMS: Source IP Group ID **[PREFIX_**SrcIPGroupID]	The IP Group to which the incoming IP call belongs. Typically, the IP Group of an incoming INVITE is determined according to the 'Inbound IP Routing Table'. **Notes:** ▪ This parameter is applicable only for IP-to-IP routing. ▪ To denote all IP Groups, leave this field empty. ▪ If this IP Group has a Serving IP Group, then all calls from this IP Group are sent to the Serving IP Group. In such a scenario, this routing table is used only if the parameter PreferRouteTable is set to 1.
Web: Src. Host Prefix EMS: Source Host Prefix **[PREFIX_**SrcHostPrefix]	The prefix of the SIP URI host name in the From header of the incoming SIP INVITE message. If this routing rule is not required, leave the field empty. **Note:** To denote any prefix, use the asterisk (*) symbol.
Web: Dest. Host Prefix EMS: Destination Host Prefix **[PREFIX_**DestHostPrefix]	The request SIP URI host name prefix of the incoming SIP INVITE message. If this routing rule is not required, leave the field empty. **Note:** To denote any prefix, use the asterisk (*) symbol.
Web: Src. Trunk Group ID EMS: Source Trunk Group ID **[PREFIX_SrcTrunkGroupID]**	The Trunk Group to which the received call belongs. The range is 1-99. **Notes:** ▪ For IP-to-IP calls, this parameter is not required. ▪ To denote any Trunk Group, enter an asterisk (*) symbol.
Web: Dest. Phone Prefix EMS: Destination Phone Prefix **[PREFIX_DestinationPrefix]**	Prefix of the called telephone number. The prefix can include up to 50 digits. **Note:** To denote any prefix, enter an asterisk (*) symbol. The prefix can be a single digit or a range of digits. For available notations, refer to "Dialing Plan Notation for Routing and Manipulation" on page 485.
Web/EMS: Source Phone Prefix **[PREFIX_SourcePrefix]**	Prefix of the calling telephone number. The prefix can include up to 50 digits. **Note:** To denote any prefix, enter an asterisk (*) symbol. The prefix can be a single digit or a range of digits. For available notations, refer to "Dialing Plan Notation for Routing and Manipulation" on page 485.

Parameter	Description
All calls matching all or any combination of the above characteristics are sent to the destination IP address defined below. **Note:** For alternative routing, additional entries of the same prefix can be configured.	
Web: Dest. IP Address EMS: Address **[PREFIX_DestAddress]**	Destination IP address (in dotted-decimal notation or FQDN) to where the call must be sent. If an FQDN is used (e.g., domain.com), DNS resolution is performed according to the parameter DNSQueryType. **Notes:** ▪ If you defined a destination IP Group (above), then this IP address is not used for routing and therefore, not required. ▪ To discard these calls, enter 0.0.0.0. For example, if you want to prohibit dialing of International calls, then in the 'Dest Phone Prefix' field, enter 00 and in the 'Dest IP Address' field, enter 0.0.0.0. ▪ For routing calls between phones connected to the device (i.e., local routing), enter the device's IP address. ▪ When the device's IP address is unknown (e.g., when DHCP is used), enter IP address 127.0.0.1. ▪ When using domain names, you must enter the DNS server's IP address or alternatively, define these names in the 'Internal DNS Table' (refer to "Configuring the Internal DNS Table" on page 167). ▪ If the string 'ENUM' is specified for the destination IP address, an ENUM query containing the destination phone number is sent to the DNS server. The ENUM reply includes a SIP URI used as the Request-URI in the outgoing INVITE and for routing (if a proxy is not used). ▪ The IP address can include the following wildcards: ✓ "x": represents single digits. For example, 10.8.8.xx depicts all addresses between 10.8.8.10 and 10.8.8.99. ✓ "*": represents any number between 0 and 255. For example, 10.8.8.* depicts all addresses between 10.8.8.0 and 10.8.8.255.
Web: Port EMS: Destination Port **[PREFIX_DestPort]**	The destination port to where you want to route the call.
Web/EMS: Transport Type **[PREFIX_TransportType]**	The transport layer type used for sending the IP calls: ▪ **[-1]** Not Configured ▪ **[0]** UDP ▪ **[1]** TCP ▪ **[2]** TLS **Note:** When set to Not Configured (-1),, the transport type defined by the parameter SIPTransportType is used.
Web: Dest IP Group ID EMS: Destination IP Group ID **[PREFIX_DestIPGroupID]**	The IP Group (1-9) to where you want to route the call. The SIP INVITE message is sent to the IP address defined for the Proxy Set ID associated with the selected IP Group. **Notes:** ▪ If you choose an IP Group, you do not need to configure a

Parameter	Description
	destination IP address. However, if both parameters are configured in this table, the INVITE message is sent only to the IP Group (and not the defined IP address). ▪ If the destination IP Group is of type USER, the device searches for a match between the Request URI (of the received INVITE) to an AOR registration record in the device's database. The INVITE is then sent to the IP address of the registered contact. ▪ If the parameter AlwaysUseRouteTable is set to 1 (refer to "Configuring the IP Groups" on page 113), then the Request URI host name in the INVITE message is set to the value defined for the parameter 'Dest. IP Address' (above); otherwise, if no IP address is defined, it is set to the value of the parameter 'SIP Group Name' (defined in the 'IP Group' table). ▪ This parameter is used as the 'Serving IP Group' in the 'Account' table for acquiring authentication user/password for this call. ▪ For defining Proxy Set ID's, refer to "Configuring the Proxy Sets Table" on page 123.
IP Profile ID **[PREFIX_ProfileId]**	IP Profile ID (defined by the parameter IPProfile) assigned to this IP destination call. This allows you to assign numerous configuration attributes (e.g., voice codes) per routing rule.
Status	Read-only field displaying the Quality of Service of the destination IP address: ▪ n/a = Alternative Routing feature is disabled. ▪ OK = IP route is available. ▪ Ping Error = No ping to IP destination; route is unavailable. ▪ QoS Low = Poor QoS of IP destination; route is unavailable. ▪ DNS Error = No DNS resolution (only when domain name is used instead of an IP address).
Web/EMS: Charge Code **[PREFIX_**MeteringCode]	Optional Charge Code (1 to 25) assigned to the routing rule. For configuring Charge Codes, refer to Configuring the Charge Codes Table on page 139. **Note:** This parameter is applicable only to FXS interfaces.

3.3.5.10.4 Configuring the Inbound IP Routing Table

The 'Inbound IP Routing Table' page allows you to configure up to 24 inbound call routing rules. The device uses these rules for the following:

- For IP-to-IP routing: identifying IP-to-IP calls and assigning them to IP Groups (referred to as Source IP Groups). These IP-to-IP calls, now pertaining to an IP Group, can later be routed to an outbound destination IP Group (refer to Configuring the Outbound IP Routing Table).
- For IP-to-Tel routing: routing incoming IP calls to Trunk Groups. The specific channel pertaining to the Trunk Group to which the call is routed is determined according to the Trunk Group's channel selection mode. The channel selection mode can be defined per Trunk Group (refer to "Configuring Trunk Group Settings" on page 104), or for allTrunk Groups using the global parameter ChannelSelectMode.

This table provides two main areas for defining a routing rule:

- **Matching Characteristics:** user-defined characteristics of the incoming IP call are defined in this area. If the characteristics match a table entry, the rule is used to route the call. One or more characteristics can be defined for the rule such as source/destination Request URI host name prefix, source (calling)/destination (called) telephone number prefix, and source IP address (from where call received).
- **Destination:** user-defined destination. If the call matches the characteristics, the device routes the call to this destination. The destination is a selected Trunk Group or a Source IP Group for IP-to-IP routing.

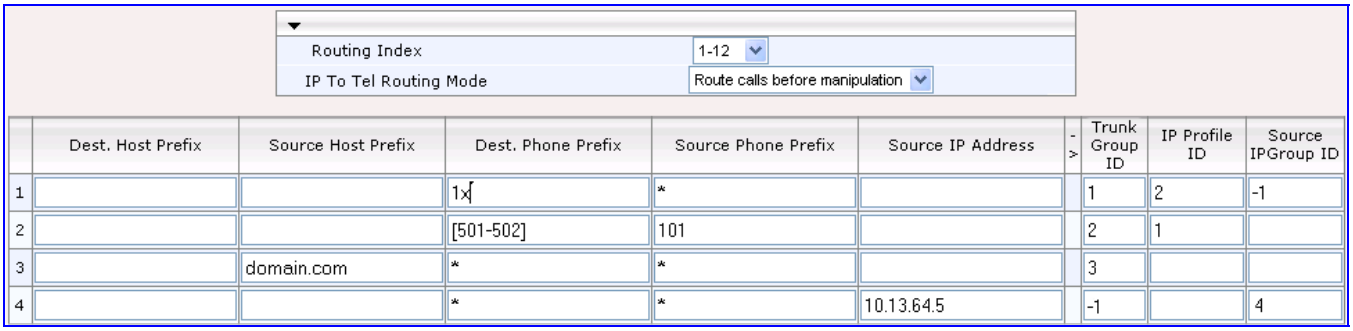
Notes:

- When a call release reason (defined in "Configuring Reasons for Alternative Routing" on page 156) is received for a specific IP-to-Tel call, an alternative Trunk Group for that call can be configured. This is done by configuring an additional routing rule for the same call characteristics, but with a different Trunk Group ID.
- You can also configure the 'Inbound IP Routing Table' using the *ini* file table parameter PSTNPrefix (refer to "Number Manipulation and Routing Parameters" on page 428).

➢ **To configure inbound IP routing rules:**

1. Open the 'Inbound IP Routing Table' page (**Configuration** tab > **Protocol Configuration** menu > **Routing Tables** submenu > **IP to Trunk Group Routing** page item).

Figure 3-94: Inbound IP Routing Table



The figure above shows the following configured inbound IP routing rules:

- **Rule 1:** If the incoming IP call destination phone prefix is between 10 and 19, the call is assigned settings configured for IP Profile ID 2 and routed to Trunk Group ID 1.
- **Rule 2:** If the incoming IP call destination phone prefix is between 501 and 502, and source phone prefix is 101, the call is assigned settings configured for IP Profile ID 1 and routed to Trunk Group ID 2.
- **Rule 3:** If the incoming IP call has a From URI host prefix as domain.com, the call is routed to Trunk Group ID 3.
- Rule 4: If the incoming IP call has IP address 10.13.64.5 in the INVITE's Contact header, the call is considered an IP-to-IP call and assigned to Source IP Group 4. This call is later routed according to the outbound IP routing rules for this Source IP Group configured in the 'Outbound IP Routing Table'.

2. From the 'Routing Index' drop-down list, select the range of entries that you want to add.
3. Configure the inbound IP routing rule according to the table below.
4. Click the **Submit** button to save your changes.
5. To save the changes so they are available after a power failure, refer to "Saving Configuration" on page 199.

Table 3-30: inbound IP Routing Table Description

Parameter	Description
IP to Tel Routing Mode **[RouteModeIP2Tel]**	Determines whether to route the incoming IP calls before or after manipulation of destination number (configured in "Configuring the Number Manipulation Tables" on page 143). ▪ **[0]** Route calls before manipulation = Incoming IP calls are routed before the number manipulation rules are applied (default). ▪ **[1]** Route calls after manipulation = Incoming IP calls are routed after the number manipulation rules are applied.
Dest. Host Prefix **[PstnPrefix_DestHostPrefix]**	The Request-URI host name prefix of the incoming SIP INVITE message. If this routing rule is not required, leave the field empty. **Note:** The asterisk (*) wildcard can be used to depict any prefix.
Source Host Prefix **[PstnPrefix_SrcHostPrefix]**	The From URI host name prefix of the incoming SIP INVITE message. If this routing rule is not required, leave the field empty. **Notes:** ▪ The asterisk (*) wildcard can be used to depict any prefix. ▪ If the P-Asserted-Identity header is present in the incoming INVITE message, then the value of this parameter is compared to the P-Asserted-Identity URI host name (and not the From header).
Dest. Phone Prefix **[PstnPrefix_DestPrefix]**	The called telephone number prefix. The prefix can include up to 49 digits. **Note:** The prefix can be a single digit or a range of digits. For available notations, refer to "Dialing Plan Notation for Routing and Manipulation" on page 485.
Source Phone Prefix **[PstnPrefix_SourcePrefix]**	The calling telephone number prefix. The prefix can include up to 49 digits. **Note:** The prefix can be a single digit or a range of digits. For available notations, refer to "Dialing Plan Notation for Routing and Manipulation" on page 485.

Parameter	Description
Source IP Address **[PstnPrefix_SourceAddress]**	The source IP address of an IP-to-Tel call (obtained from the Contact header in the INVITE message) that can be used for routing decisions. **Notes:** ▪ You can configure from where the source IP address is obtained, using the parameter SourceIPAddressInput. ▪ The source IP address can include the following wildcards: ✓ "x": depicts single digits. For example, 10.8.8.xx represents all the addresses between 10.8.8.10 and 10.8.8.99. ✓ "*": depicts any number between 0 and 255. For example, 10.8.8.* represents all addresses between 10.8.8.0 and 10.8.8.255.
Calls matching all or any combination of the above characteristics are sent to the Trunk Group ID or assigned to the source IP Group for IP-to-IP routing defined below. **Note:** For alternative routing, additional entries of the same characteristics can be configured.	
Trunk Group ID **[PstnPrefix_TrunkGroupId]**	For IP-to-Tel calls: The Trunk Group to which the incoming SIP call is assigned if it matches all or any combination of the parameters described above. For IP-to-IP calls: Identifies the call as an IP-to-IP call when this parameter is set to -1.
IP Profile ID **[PstnPrefix_ProfileId]**	The IP Profile (configured in "Configuring P Profiles" on page 133) to assign to the inbound IP call.
Source IP Group ID **[PstnPrefix_SrcIPGroupID]**	For IP-to-Tel calls: The source IP Group associated with the incoming IP-to-Tel call. This is the IP Group from where the INVITE message originated. This IP Group can later be used as the 'Serving IP Group' in the Account table for obtaining authentication user name/password for this call (refer to "Configuring the Account Table" on page 118). For IP-to-IP calls: The IP Group you want to assign the incoming IP call. This IP Group can later be used for outbound IP routing and as the 'Serving IP Group' in the Account table for obtaining authentication user name/password for this call (refer to Configuring the Account Table on page 118).

3.3.5.10.5 Configuring the Internal DNS Table

The 'Internal DNS Table' page, similar to a DNS resolution is used to translate up to 20 host (domain) names into IP addresses (e.g., when using the 'Outbound IP Routing Table'). Up to four different IP addresses can be assigned to the same host name, typically used for alternative routing (for Tel-to-IP call routing).

Notes:

- The device initially attempts to resolve a domain name using the Internal DNS table. If the domain name isn't listed in the table, the device performs a DNS resolution using an external DNS server.
- You can also configure the DNS table using the *ini* file table parameter DNS2IP (refer to "DNS Parameters" on page 257).

➢ **To configure the internal DNS table:**

1. Open the 'Internal DNS Table' page (**Configuration** tab > **Protocol Configuration** menu > **Routing Tables** submenu > **Internal DNS Table** page item).

Figure 3-95: Internal DNS Table Page

	Domain Name	First IP Address	Second IP Address	Third IP Address	Fourth IP Address
1	DomainName.com	10.8.2.15	10.8.4.20	10.8.6.17	10.8.6.18
2					
3					
4					

2. In the 'Domain Name' field, enter the host name to be translated. You can enter a string of up to 31 characters long.
3. In the 'First IP Address' field, enter the first IP address (in dotted-decimal format notation) to which the host name is translated.
4. Optionally, in the 'Second IP Address', 'Third IP Address', and 'Second IP Address' fields, enter the next IP addresses to which the host name is translated.
5. Click the **Submit** button to save your changes.
6. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

3.3.5.10.6 Configuring the Internal SRV Table

The 'Internal SRV Table' page provides a table for resolving host names to DNS A-Records. Three different A-Records can be assigned to each host name. Each A-Record contains the host name, priority, weight, and port.

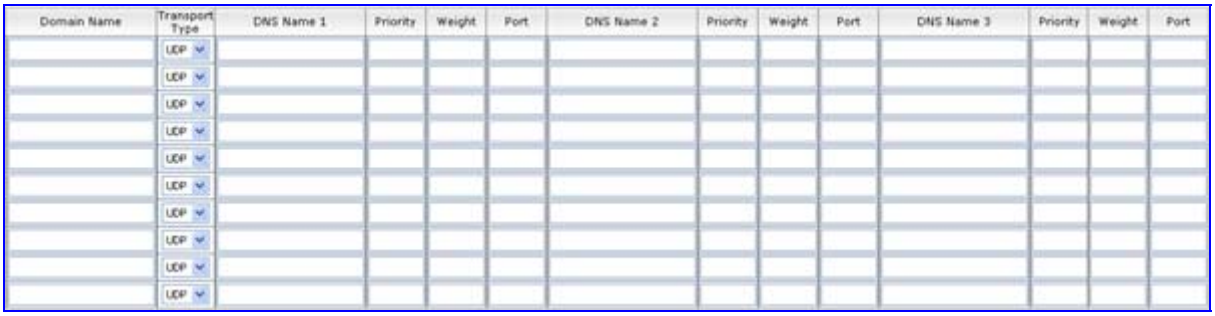
Notes:

- If the Internal SRV table is configured, the device initially attempts to resolve a domain name using this table. If the domain name isn't found, the device performs an Service Record (SRV) resolution using an external DNS server.
- You can also configure the Internal SRV table using the *ini* file table parameter SRV2IP (refer to "DNS Parameters" on page 257).

➢ **To configure the Internal SRV table:**

1. Open the 'Internal SRV Table' page (**Configuration** tab > **Protocol Configuration** menu > **Routing Tables** submenu > **Internal SRV Table** page item).

Figure 3-96: Internal SRV Table Page



2. In the 'Domain Name' field, enter the host name to be translated. You can enter a string of up to 31 characters long.
3. From the 'Transport Type' drop-down list, select a transport type.
4. In the 'DNS Name 1' field, enter the first DNS A-Record to which the host name is translated.
5. In the 'Priority', 'Weight' and 'Port' fields, enter the relevant values
6. Repeat steps 4 through 5, for the second and third DNS names, if required.
7. Repeat steps 2 through 6, for each entry.
8. Click the **Submit** button to save your changes.
9. To save the changes so they are available after a hardware reset or power fail, refer to "Saving Configuration" on page 199.

3.3.5.10.7 Configuring Release Cause Mapping

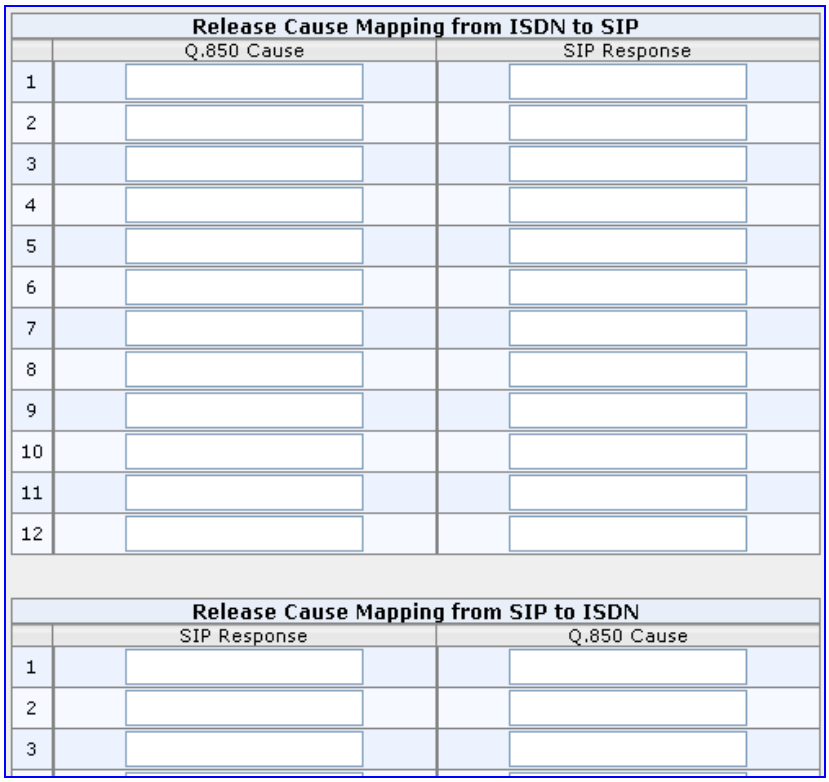
The 'Release Cause Mapping' page consists of two groups that allow the device to map up to 12 different SIP Response Codes to Q.850 Release Causes and vice versa, thereby overriding the hard-coded mapping mechanism (described in "Release Reason Mapping" on page 626).

Note: You can also configure SIP Responses-Q.850 Release Causes mapping using the *ini* file table parameters CauseMapISDN2SIP and CauseMapSIP2ISDN (refer to "ISDN and CAS Interworking-Related Parameters" on page 392).

➢ **To configure Release Cause Mapping:**

1. Open the 'Release Cause Mapping' page (**Configuration** tab > **Protocol Configuration** menu > **Routing Tables** submenu > **Release Cause Mapping** page item).

Figure 3-97: Release Cause Mapping Page



2. In the 'Release Cause Mapping from ISDN to SIP' group, map different Q.850 Release Causes to SIP Responses.
3. In the 'Release Cause Mapping from SIP to ISDN' group, map different SIP Responses to Q.850 Release Causes.
4. Click the **Submit** button to save your changes.
5. To save the changes so they are available after a power fail, refer to "Saving Configuration" on page 199.

3.3.5.10.8 Configuring Call Forward upon Busy Trunk

The 'Forward on Busy Trunk Destination' page allows you to configure forwarding of IP-to-Tel calls to a different (alternative) IP destination, using SIP 3xx response, upon the following scenarios:

- For digital PSTN interfaces: If a Trunk Group has no free channels (i.e., “busy” Trunk Group).
- For analog interfaces: if an unavailable FXS/FXO Trunk Group exists. This feature can be used, for example, to forward the call to another FXS/FXO device.

The alternative destination (i.e., IP address, port and transport type) is configured per Trunk Group.

The device forwards calls using this table only if no alternative IP-to-Tel routing has been configured or alternative routing fails, and one of the following reasons (included in the SIP Diversion header of 3xx messages) exists:

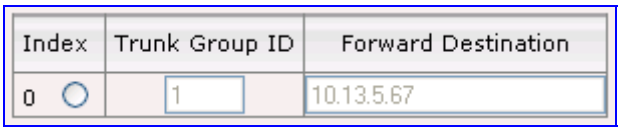
- For digital PSTN interfaces: “out-of-service” - all trunks are unavailable/disconnected
- "unavailable":
  - For digital PSTN interfaces: All trunks are busy or unavailable
  - For analog interfaces: All FXS/FXO lines pertaining to a Trunk Group are busy or unavailable

Note: You can also configure the Forward on Busy Trunk Destination table using the *ini* file parameter table ForwardOnBusyTrunkDest.

➢ **To configure the Forward on Busy Trunk Destination table:**

1. Open the 'Forward on Busy Trunk Destination' page (**Configuration** tab > **Protocol Configuration** menu > **Routing Tables** submenu > **Forward on Busy Trunk Dest** page item).

Figure 3-98: Forward on Busy Trunk Destination Page



The figure above includes a configuration entry that forwards IP-to-Tel calls destined for Trunk Group ID 2 to destination IP address 10.13.5.67.

2. Click the **Submit** button to save your changes.
3. To save the changes so they are available after a power fail, refer to "Saving Configuration" on page 199.

3.3.5.11 Endpoint Settings

The **Endpoint Settings** submenu allows you to configure analog (FXS/FXO) port-specific parameters. This submenu includes the following page items:

- Authentication (refer to "Configuring Authentication" on page 171)
- Automatic Dialing (refer to "Configuring Automatic Dialing" on page 172)
- Caller Display Information (refer to "Configuring Caller Display Information" on page 173)
- Call Forward (refer to "Configuring Call Forward" on page 175)
- Caller ID Permissions (refer to "Configuring Caller ID Permissions" on page 176)
- Call Waiting (refer to "Configuring Call Waiting" on page 177)

Note: The Endpoint Settings menu appears only if the device is installed with an FXS or FXO module.

3.3.5.11.1 Configuring Authentication

The 'Authentication' page defines a user name and password for authenticating each device port. Authentication is typically used for FXS interfaces, but can also be used for FXO interfaces.

Notes:

- For configuring whether authentication is performed per port or for the entire device, use the parameter AuthenticationMode. If authentication is for the entire device, the configuration on this page is ignored.
- If either the user name or password fields are omitted, the port's phone number and global password (using the Password parameter) are used instead.
- After you click the **Submit** button, the password is displayed as an asterisk (*).
- You can also configure Authentication using the *ini* file table parameter Authentication (refer to "SIP Configuration Parameters" on page 284).

➢ **To configure the Authentication Table:**

1. Set the parameter 'Authentication Mode' (AuthenticationMode) to 'Per Endpoint'.
2. Open the 'Authentication' page (**Configuration** tab > **Protocol Configuration** menu > **Endpoint Settings** submenu > **Authentication** page item).

Figure 3-99: Authentication Page

Gateway Port	User Name	Password
Module 1 Port 1 FXS	user1	•••••••
Module 1 Port 2 FXS	user2	•••••••
Module 2 Port 1 FXO		
Module 2 Port 2 FXO		
Module 2 Port 3 FXO		
Module 2 Port 4 FXO		

3. In the 'User Name' and 'Password' fields corresponding to a port, enter the user name and password respectively.
4. Click the **Submit** button to save your changes.
5. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

3.3.5.11.2 Configuring Automatic Dialing

The 'Automatic Dialing' page allows you to define a telephone number that is automatically dialed when an FXS or FXO port is used (e.g., off-hooked).

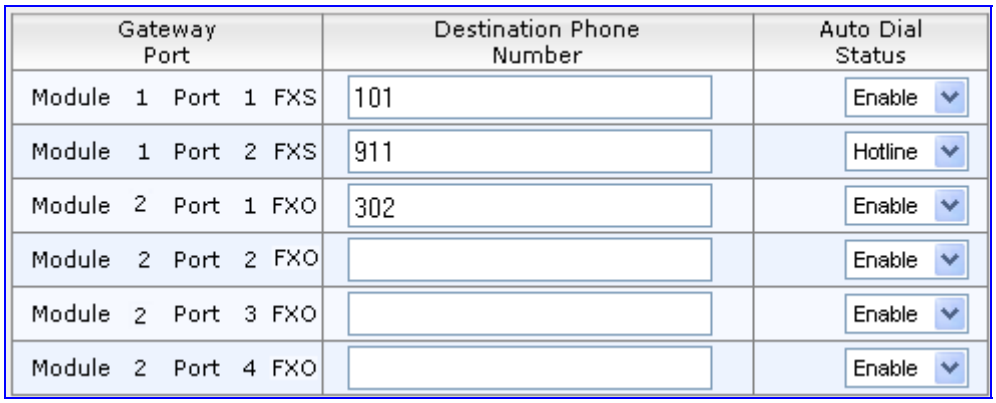
Notes:

- After a ring signal is detected on an 'Enabled' FXO port, the device initiates a call to the destination number without seizing the line. The line is seized only after the call is answered.
- After a ring signal is detected on a 'Disabled' or 'Hotline' FXO port, the device seizes the line.
- You can also configure automatic dialing using the *ini* file table parameter TargetOfChannel.
- You can configure the device to play a Busy/Reorder tone to the Tel side upon receiving a SIP 4xx, 5xx, or 6xx response from the IP side (i.e., Tel-to-IP call failure), using the *ini* file parameter FXOAutoDialPlayBusyTone (refer to "SIP Configuration Parameters" on page 284).

➢ **To configure Automatic Dialing:**

1. Open the 'Automatic Dialing' page (**Configuration** tab > **Protocol Configuration** menu > **Endpoint Settings** submenu > **Automatic Dialing** page item).

Figure 3-100: Automatic Dialing Page



2. In the 'Destination Phone Number' field corresponding to a port, enter the telephone number that you want automatically dialed.
3. From the 'Auto Dial Status' drop-down list, select one of the following:
   - Enable **[1]**: The number in the 'Destination Phone Number' field is automatically dialed if the phone is off-hooked (for FXS interfaces) or a ring signal (from PBX/PSTN switch) is detected (FXO interfaces). The FXO line is seized only after the SIP call is answered.
   - Disable **[0]**: The automatic dialing feature for the specific port is disabled (i.e., the number in the 'Destination Phone Number' field is ignored).
   - Hotline **[2]**:
     - **FXS interfaces:** When a phone is off-hooked and no digit is dialed for a user-defined time (configured using the parameter HotLineToneDuration), the number in the 'Destination Phone Number' field is automatically dialed.
     - **FXO interfaces:** If a ring signal is detected, the device seizes the FXO line, plays a dial tone, and then waits for DTMF digits. If no digits are detected for a user-defined time (configured using the parameter HotLineToneDuration), the number in the 'Destination Phone Number' field is automatically dialed by sending a SIP INVITE message with this number.
4. Click the **Submit** button to save your changes.
5. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

3.3.5.11.3 Configuring Caller Display Information

The 'Caller Display Information' page allows you to enable the device to send Caller ID information to IP when a call is made. The called party can use this information for caller identification. The information configured in this page is sent in an INVITE message in the 'From' header. For information on Caller ID restriction according to destination/source prefixes, refer to "Configuring the Number Manipulation Tables" on page 143.

➢ **To configure the Caller Display Information:**

1. Open the 'Caller Display Information' page (**Configuration** tab > **Protocol Configuration** menu > **Endpoint Settings** submenu > **Caller Display Information** page item).

Figure 3-101: Caller Display Information Page

Gateway Port	Caller ID/Name	Presentation
Module 1 Port 1 FXS	Private	Restricted
Module 1 Port 2 FXS	Susan C.	Restricted
Module 2 Port 1 FXO	Lee P.	Allowed
Module 2 Port 2 FXO	Ronaldo	Allowed
Module 2 Port 3 FXO		Allowed
Module 2 Port 4 FXO		Allowed

2. In the' Caller ID/Name' field corresponding to the desired port, enter the Caller ID string (up to 18 characters).
3. From the 'Presentation' drop-down list, select one of the following:
   - 'Allowed' **[0]** - sends the string defined in the 'Caller ID/Name' field when a Tel-to-IP call is made using the corresponding device port.
   - 'Restricted' **[1]** - the string defined in the 'Caller ID/Name' field is not sent.
4. Click the **Submit** button to save your changes.
5. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

Notes:

- When FXS ports receive 'Private' or 'Anonymous' strings in the From header, they don't send the calling name or number to the Caller ID display.
- If Caller ID name is detected on an FXO line (EnableCallerID = 1), it is used instead of the Caller ID name defined on this page.
- When the 'Presentation' field is set to 'Restricted', the Caller ID is sent to the remote side using only the P-Asserted-Identity and P-Preferred-Identity headers (AssertedIdMode).
- The value of the 'Presentation' field can be overridden by configuring the 'Presentation' field in the 'Source Number Manipulation' table (refer to "Configuring the Number Manipulation Tables" on page 143).
- You can also configure the Caller Display Information table using the *ini* file table parameter CallerDisplayInfo.

3.3.5.11.4 Configuring Call Forward

The 'Call Forwarding Table' page allows you to forward (redirect) IP-to-Tel calls (using SIP 302 response) originally destined to specific device ports, to other device ports or to an IP destination.

Notes:

- Ensure that the Call Forward feature is enabled (default) for the settings on this page to take effect. To enable Call Forward, use the parameter EnableForward ("Configuring Supplementary Services" on page 138).
- You can also configure the Call Forward table using the *ini* file table parameter FwdInfo.

➢ **To configure Call Forward per port:**

1. Open the 'Call Forward Table' page (**Configuration** tab > **Protocol Configuration** menu > **Endpoint Settings** submenu > **Call Forward** page item).

Figure 3-102: Call Forward Table Page

Gateway Port	Forward Type	Forward to Phone Number	Time for No Reply Forward
Module 1 Port 1 FXS	On busy	201	30
Module 1 Port 2 FXS	Unconditional	202@10.2.1.1	30
Module 2 Port 1 FXO	No Answer	203	30
Module 2 Port 2 FXO	Deactivate		30
Module 2 Port 3 FXO	Deactivate		30
Module 2 Port 4 FXO	Deactivate		30

2. Configure the Call Forward parameters for each port according to the table below.
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

Table 3-31: Call Forward Table

Parameter	Description
Forward Type	Determines the scenario for forwarding a call. ▪ **[0]** Deactivate = Don't forward incoming calls (default). ▪ **[1]** On Busy = Forward incoming calls when the port is busy. ▪ **[2]** Unconditional = Always forward incoming calls. ▪ **[3]** No Answer = Forward incoming calls that are not answered within the time specified in the 'Time for No Reply Forward' field. ▪ **[4]** On Busy or No Answer = Forward incoming calls when the port is busy or when calls are not answered within the time specified in the 'Time for No Reply Forward' field. ▪ **[5]** Do Not Disturb = Immediately reject incoming calls.

Parameter	Description
Forward to Phone Number	The telephone number or URI (<number>@<IP address>) to where the call is forwarded. **Note:** If this field only contains a telephone number and a Proxy isn't used, the 'forward to' phone number must be specified in the 'Outbound IP Routing Table' (refer to "Configuring the Outbound IP Routing Table" on page 158).
Time for No Reply Forward	If you have set the 'Forward Type' for this port to 'No Answer', enter the number of seconds the device waits before forwarding the call to the phone number specified.

3.3.5.11.5 Configuring Caller ID Permissions

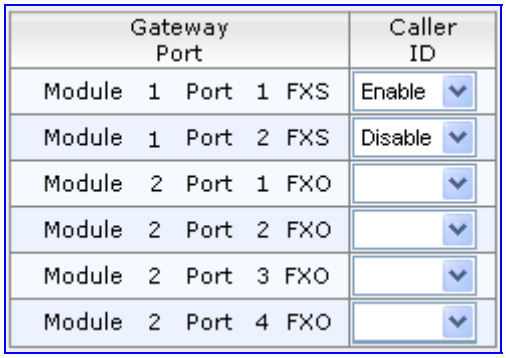
The 'Caller ID Permissions' page allows you to enable or disable (per port) the Caller ID generation (for FXS interfaces) and detection (for FXO interfaces). If a port isn't configured, its Caller ID generation/detection are determined according to the global parameter EnableCallerID described in "Configuring Supplementary Services" on page 138.

Note: You can also configure the Caller ID Permissions table using the *ini* file table parameter EnableCallerID.

➢ **To configure Caller ID Permissions per port:**

1. Open the 'Caller ID Permissions' page (**Configuration** tab > **Protocol Configuration** menu > **Endpoint Settings** submenu > **Caller ID Permissions** page item).

Figure 3-103: Caller ID Permissions Page



2. From the 'Caller ID' drop-down list, select one of the following:
   - 'Enable': Enables Caller ID generation (FXS) or detection (FXO) for the specific port.
   - 'Disable': Caller ID generation (FXS) or detection (FXO) for the specific port is disabled.
   - Not defined: Caller ID generation (FXS) or detection (FXO) for the specific port is determined according to the parameter 'Enable Caller ID' (described in "Configuring Supplementary Services" on page 138).
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

3.3.5.11.6 Configuring Call Waiting

The 'Call Waiting' page allows you to enable or disable call waiting per device FXS port.

Notes:

- This page is applicable only to FXS interfaces.
- Instead of using this page, you can enable or disable call waiting for all the device's ports, using the global call waiting parameter 'Enable Call Waiting' (refer to "Configuring Supplementary Services" on page 138).
- You can also configure the Call Waiting table using the *ini* file table parameter CallWaitingPerPort (refer to "SIP Configuration Parameters" on page 284).
- For additional call waiting configuration, refer to the following parameters: FirstCallWaitingToneID (in the CPT file), TimeBeforeWaitingIndication, WaitingBeepDuration, TimeBetweenWaitingIndications, and NumberOfWaitingIndications.

➢ **To configure Call Waiting:**

1. Open the 'Caller Waiting' page (**Configuration** tab > **Protocol Configuration** menu > **Endpoint Settings** submenu > **Call Waiting** page item).

Figure 3-104: Caller Waiting Page

Gateway Port	Call Waiting Configuration
Module 1 Port 1 FXS	Enable
Module 1 Port 2 FXS	Enable
Module 2 Port 1 FXO	
Module 2 Port 2 FXO	
Module 2 Port 3 FXO	
Module 2 Port 4 FXO	

2. From the 'Call Waiting Configuration' drop-down list corresponding to the port you want to configure for call waiting, select one of the following options:
   - 'Enable': Enables call waiting for the specific port. When the device receives a call on a busy endpoint (port), it responds with a 182 response (not with a 486 busy). The device plays a call waiting indication signal. When hook-flash is detected by the device, the device switches to the waiting call. The device that initiated the waiting call plays a Call Waiting Ringback tone to the calling party after a 182 response is received.
   - 'Disable': No call waiting for the specific port.
   - Empty: Call waiting is determined according to the global parameter 'Enable Call Waiting' (described in "Configuring Supplementary Services" on page 138).
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

3.3.5.12 Configuring Digital Gateway Parameters

The 'Digital Gateway Parameters' page allows you to configure miscellaneous digital parameters. For a description of these parameters, refer to "Configuration Parameters Reference" on page 247.

➢ **To configure the digital gateway parameters:**

1. Open the 'Digital Gateway Parameters' page (**Configuration** tab > **Protocol Configuration** menu > **Digital Gateway** submenu > **Digital Gateway Parameters** page item).

Figure 3-105: Digital Gateway Parameters Page

2. Configure the parameters as required.
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

3.3.5.13 Configuring the IPmedia Parameters

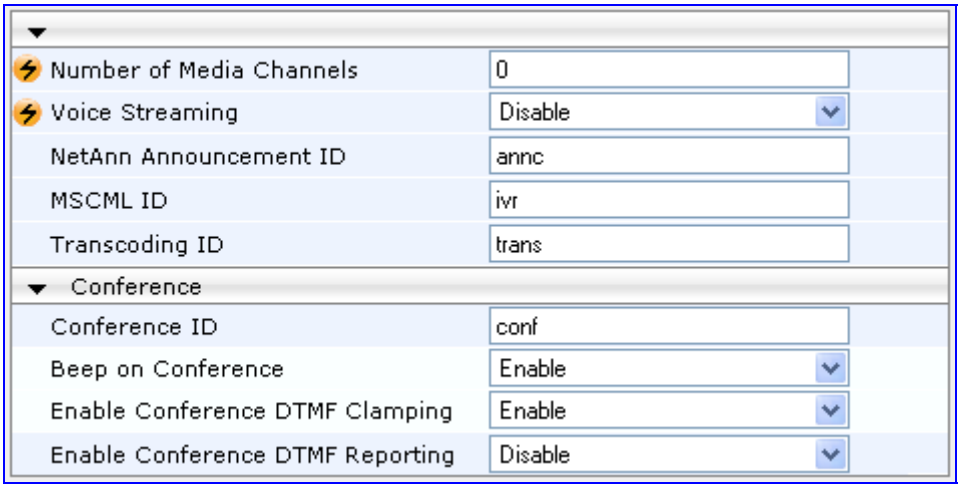
The 'IP Media Settings' page allows you to configure the IP media parameters. For a description of these parameters, refer to "Configuration Parameters Reference" on page 247.

Note: This page is applicable only to Mediant 1000. This page is Software Upgrade Key dependant (refer to "Upgrading the Software Upgrade Key" on page 202).

➢ **To configure the IP media parameters:**

1. Open the 'IP Media Settings' page (**Configuration** tab > **Protocol Configuration** menu > **IP Media** submenu > **IP Media Settings** page item).

Figure 3-106: IP Media Settings Page



2. Configure the parameters as required.
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" **on page 199.**

3.3.5.14 SAS Parameters

The **SAS** submenu allows you to configure the SAS application. This submenu includes the **Stand Alone Survivability** item page (refer to "Configuring Stand-Alone Survivability Parameters" on page 180), from which you can also access the 'IP2IP Routing Table' page for configuring SAS routing rules (refer to "Configuring the IP2IP Routing Table (SAS)" on page 181).

Notes:

- The SAS menu and its page items appear only if you have enabled the SAS application (refer to "Enabling Applications" on page 101) and the SAS application is included in the device's Software Upgrade Key (refer to "Loading a Software Upgrade Key" on page 202).
- For a detailed explanation on SAS, refer to "Stand-Alone Survivability (SAS) Feature" on page 516.

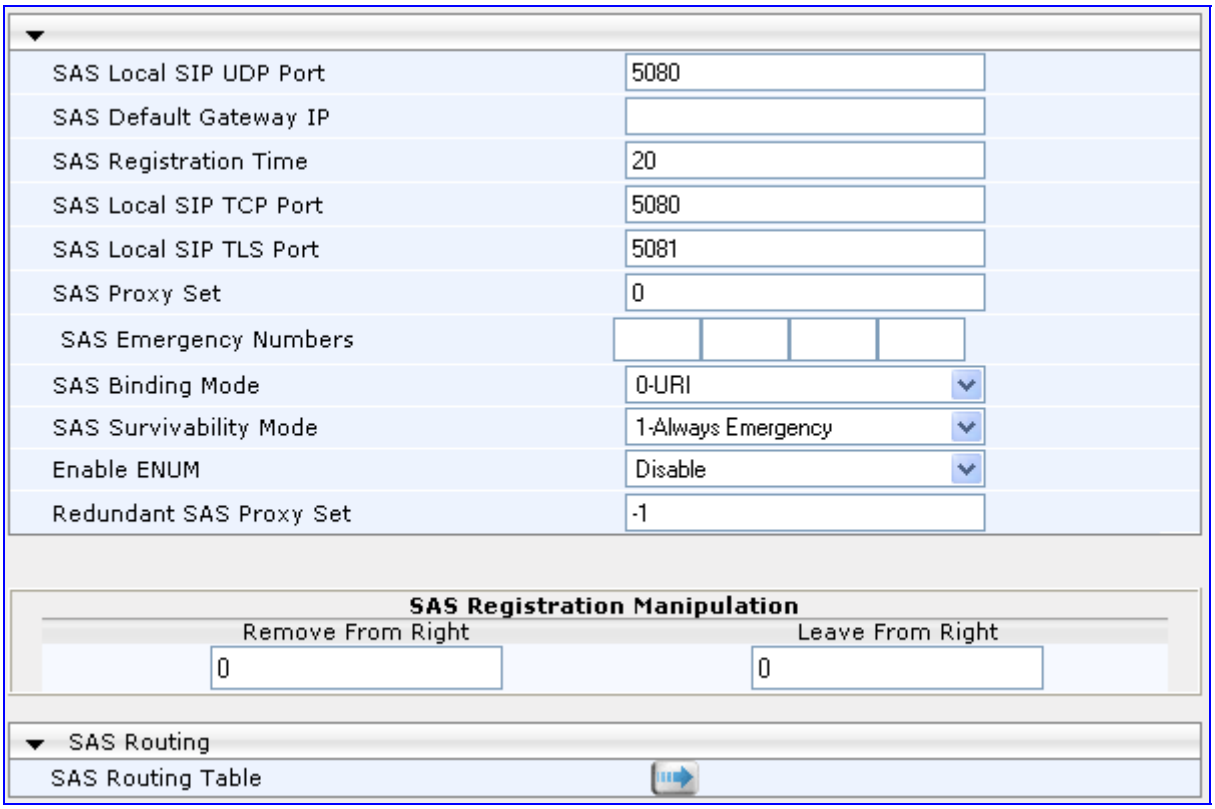
3.3.5.14.1 Configuring Stand-Alone Survivability Parameters

The 'SAS Configuration' page allows you to configure the device's Stand-Alone Survivability (SAS) feature. This feature is useful for providing a local backup through the PSTN in Small or Medium Enterprises (SME) that are serviced by IP Centrex services. In such environments, the enterprise's incoming and outgoing telephone calls (external and internal) are controlled by the Proxy, which communicates with the enterprise through the WAN interface. SAS ensures that incoming, outgoing, and internal calls service is maintained in case of WAN or Proxy failure, using a PSTN (or an alternative VoIP) backup connection and the device's internal call routing. To utilize the SAS feature, the VoIP CPEs such as IP phones or residential gateways need to be defined so that their Proxy and Registrar destination addresses and UDP port equal the SAS feature's IP address and SAS local SIP UDP port.

➢ **To configure the Stand-Alone Survivability parameters:**

1. Open the 'SAS Configuration' page (**Configuration** tab > **Protocol Configuration** menu > **SAS** submenu > **Stand Alone Survivability** page item).

Figure 3-107: SAS Configuration Page



2. Configure the parameters as described in "SIP Configuration Parameters" on page 284.
3. Click the **Submit** button to apply your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

To configure the SAS Routing table, under the **SAS Routing** group, click the **SAS Routing Table** button to open the 'IP2IP Routing Table' page. For a description of this table, refer to "Configuring the IP2IP Routing Table (SAS)" on page 181.

3.3.5.14.2 Configuring the IP2IP Routing Table (SAS)

The 'IP2IP Routing Table' page allows you to configure up to 120 SAS routing rules (for Normal and Emergency modes). The device routes the SAS call (received SIP INVITE message) once a rule in this table is matched. If the characteristics of an incoming call do not match the first rule, the call characteristics is then compared to the settings of the second rule, and so on until a matching rule is located. If no rule is matched, the call is rejected.

When SAS receives a SIP INVITE request from a proxy server, the following routing logic is performed:

a. Sends the request according to rules configured in the IP2IP Routing table.

b. If no matching routing rule exists, the device sends the request according to its SAS registration database.

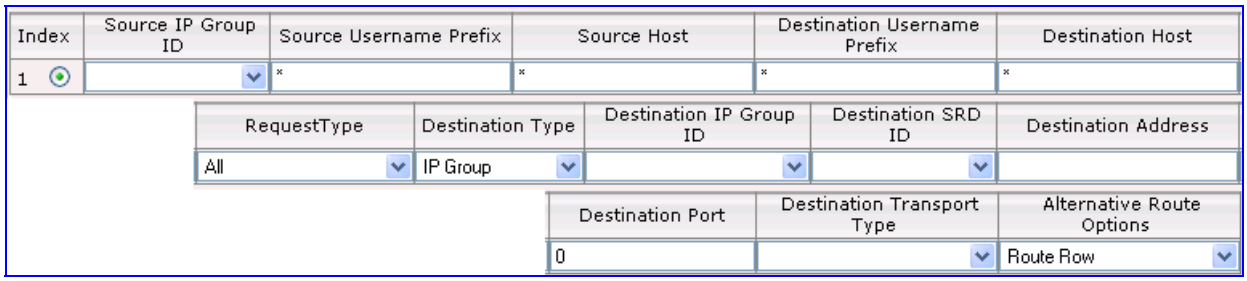
c. If no routing rule is located in the database, the device sends the request according to the Request-URI header.

Note: The IP2IP Routing table can also be configured using the *ini* file table parameter IP2IPRouting (refer to "SIP Configuration Parameters" on page 284).

➢ **To configure the IP2IP Routing table for SAS:**

1. In the 'SAS Configuration' page (refer to "Configuring Stand-Alone Survivability Parameters" on page 180), click the **SAS Routing Table** button; the 'IP2IP Routing Table' page appears.

Figure 3-108: IP2IP Routing Page



2. Add an entry and then configure it according to the table below.
3. Click the **Apply** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

Table 3-32: SAS Routing Table Parameters

Parameter	Description
Matching Characteristics	
Source Username Prefix **[IP2IPRouting_SrcUsernamePrefix]**	The prefix of the user part of the incoming INVITE's source URI (usually the From URI). The default is "*". **Note:** The prefix can be a single digit or a range of digits. For available notations, refer to "Dialing Plan Notation for Routing and Manipulation" on page 485.

Parameter	Description
Source Host **[IP2IPRouting_SrcHost]**	The host part of the incoming SIP INVITE's source URI (usually the From URI). If this rule is not required, leave the field empty. To denote any host name, use the asterisk (*) symbol. The default is "*".
Destination Username Prefix **[IP2IPRouting_DestUsernamePrefix]**	The prefix of the incoming SIP INVITE's destination URI (usually the Request URI) user part. If this rule is not required, leave the field empty. To denote any prefix, use the asterisk (*) symbol. The default is "*".
Destination Host **[IP2IPRouting_DestHost]**	The host part of the incoming SIP INVITE's destination URI (usually the Request URI). If this rule is not required, leave the field empty. The asterisk (*) symbol can be used to depict any destination host. The default is "*".
Operation Routing Rule (performed when match occurs in above characteristics)	
Destination Type **[IP2IPRouting_DestType]**	Determines the destination type to which the outgoing INVITE is sent. ▪ **[0]** IP Group (default) = The INVITE is sent to the IP Group's Proxy Set (if the IP Group is of SERVER type) \ registered contact from the database (if USER type). ▪ **[1]** DestAddress = The INVITE is sent to the address configured in the following fields: 'Destination Address', 'Destination Port', and 'Destination Transport Type'. ▪ **[2]** Request URI = The INVITE is sent to the address indicated in the incoming Request URI. If the fields 'Destination Port' and 'Destination Transport Type' are configured, the incoming Request URI parameters are overridden and these fields take precedence. ▪ **[3]** ENUM = An ENUM query is sent to conclude the destination address. If the fields 'Destination Port' and 'Destination Transport Type' are configured, the incoming Request URI parameters are overridden and these fields take precedence.
Destination IP Group ID **[IP2IPRouting_DestIPGroupID]**	The IP Group ID to where you want to route the call. The INVITE messages are sent to the IP address(es) defined for the Proxy Set associated with this IP Group. If you select an IP Group, it is unnecessary to configure a destination IP address (in the 'Destination Address' field). However, if both parameters are configured, the the IP Group takes precedence. If the destination IP Group is of USER type, the device searches for a match between the Request URI (of the received INVITE) to an AOR registration record in the device's database. The INVITE is then sent to the IP address of the registered contact. The default is -1. **Note:** This parameter is only relevant if the parameter 'Destination Type' is set to 'IP Group'. However, regardless of the settings of the parameter 'Destination Type', the IP Group is still used - only for determining the IP Profile.

Parameter	Description
Destination Address **[IP2IPRouting_DestAddress]**	The destination IP address (or domain name, e.g., domain.com) to where the call is sent. **Notes:** ▪ This parameter is applicable only if the parameter 'Destination Type' is set to 'Dest Address' **[1].** ▪ When using domain names, enter a DNS server IP address or alternatively, define these names in the 'Internal DNS Table' (refer to "Configuring the Internal SRV Table" on page 168).
Destination Port **[IP2IPRouting_DestPort]**	The destination port to where the call is sent.
Destination Transport Type **[IP2IPRouting_DestTransportType]**	The transport layer type for sending the call: ▪ **[-1]** Not Configured (default) ▪ **[0]** UDP ▪ **[1]** TCP ▪ **[2]** TLS **Note:** When this parameter is set to -1, the transport type is determined by the parameter SIPTransportType.

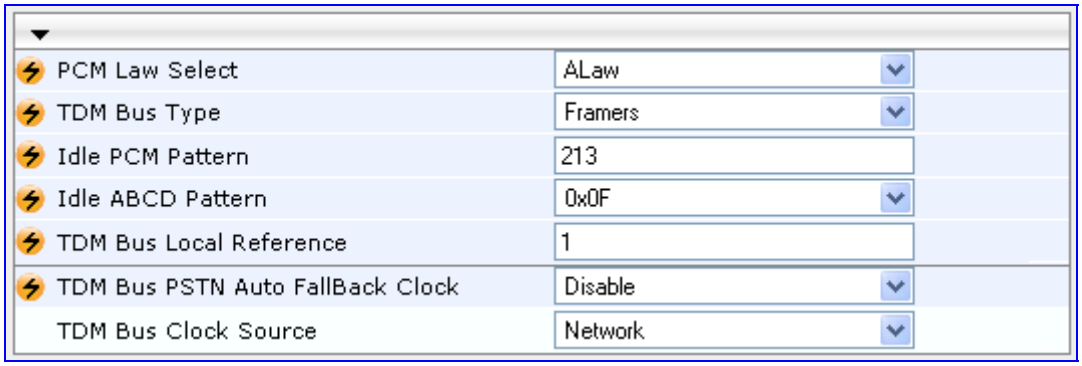
3.3.6 Configuring TDM Bus Settings

The 'TDM Bus Settings' page allows you to configure the device's Time-Division Multiplexing (TDM) bus settings. For detailed information on configuring the device's clock settings, refer to "Clock Settings" on page 625. For a description of these parameters, refer to "Configuration Parameters Reference" on page 247.

➢ **To configure the TDM Bus settings:**

1. Open the 'TDM Bus Settings' page (**Configuration** tab > **TDM Configuration** menu > **TDM Bus Settings** page item).

Figure 3-109: TDM Bus Settings Page



2. Configure the parameters as required.
3. Click the **Submit** button to save your changes.
4. Save the changes to flash memory, refer to "Saving Configuration" on page 199.

3.3.7 Advanced Applications

The **Advanced Applications** menu allows you to configure advanced SIP-based applications. This menu includes the following page items:

- Voice Mail Settings (refer to Configuring Voice Mail Parameters on page 185)
- RADIUS Parameters (refer to "Configuring RADIUS Accounting Parameters" on page 186)
- FXO Settings (refer to Configuring FXO Parameters on page 187)
- LDAP Settings (refer to Configuring LDAP Settings on page 188)

3.3.7.1 Configuring Voice Mail Parameters

The 'Voice Mail Settings' page allows you to configure the voice mail parameters. For a description of these parameters, refer to "Configuration Parameters Reference" on page 247.

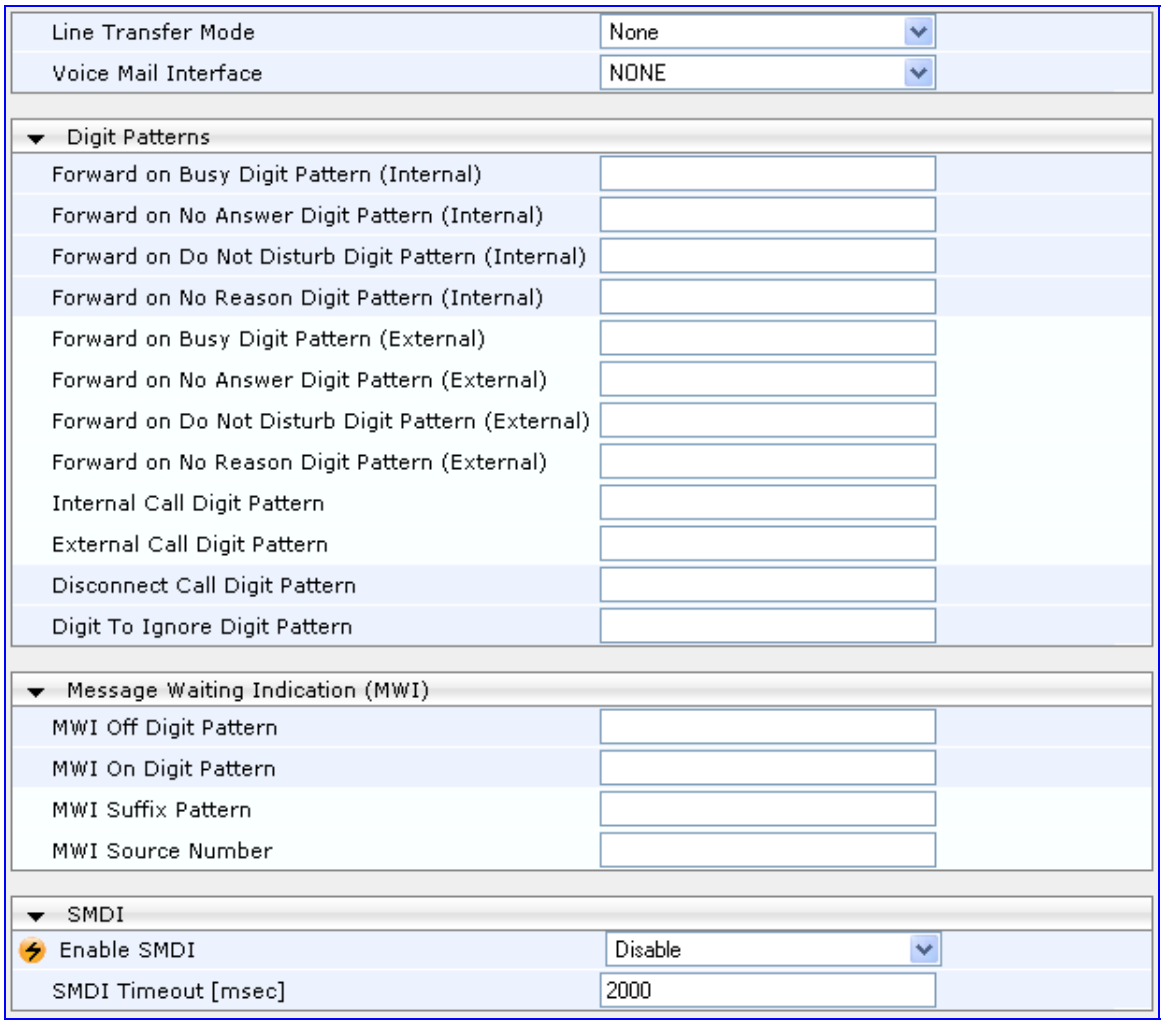
Notes:

- The 'Voice Mail Settings' page is available only for FXO and CAS interfaces.
- For detailed information on configuring the voice mail application, refer to the *CPE Configuration Guide for Voice Mail User's Manual*.

➢ **To configure the Voice Mail parameters:**

1. Open the 'Voice Mail Settings' page (**Configuration** tab > **Advanced Applications** menu > **Voice Mail Settings** page item).

Figure 3-110: Voice Mail Settings Page


Line Transfer Mode
None
Voice Mail Interface
NONE
Digit Patterns
Forward on Busy Digit Pattern (Internal)
Forward on No Answer Digit Pattern (Internal)
Forward on Do Not Disturb Digit Pattern (Internal)
Forward on No Reason Digit Pattern (Internal)
Forward on Busy Digit Pattern (External)
Forward on No Answer Digit Pattern (External)
Forward on Do Not Disturb Digit Pattern (External)
Forward on No Reason Digit Pattern (External)
Internal Call Digit Pattern
External Call Digit Pattern
Disconnect Call Digit Pattern
Digit To Ignore Digit Pattern
Message Waiting Indication (MWI)
MWI Off Digit Pattern
MWI On Digit Pattern
MWI Suffix Pattern
MWI Source Number
SMDI
Enable SMDI
Disable
SMDI Timeout [msec]
2000

2. Configure the parameters as required.
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

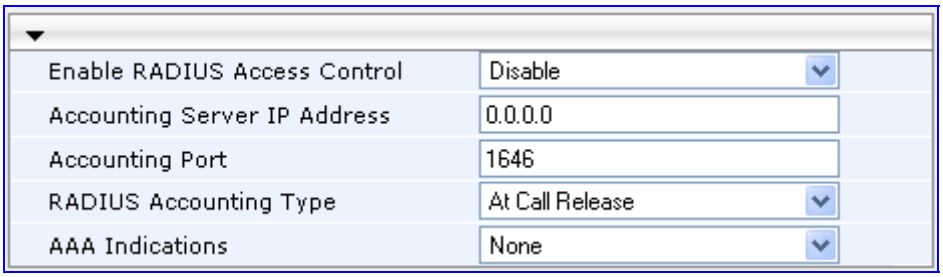
3.3.7.2 Configuring RADIUS Accounting Parameters

The 'RADIUS Parameters' page is used for configuring the Remote Authentication Dial In User Service (RADIUS) accounting parameters. For a description of these parameters, refer to "Configuration Parameters Reference" on page 247.

➢ **To configure the RADIUS parameters:**

1. Open the 'RADIUS Parameters' page (**Configuration** tab > **Advanced Applications** menu > **RADIUS Parameters** page item).

Figure 3-111: RADIUS Parameters Page



2. Configure the parameters as required.
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

3.3.7.3 Configuring FXO Parameters

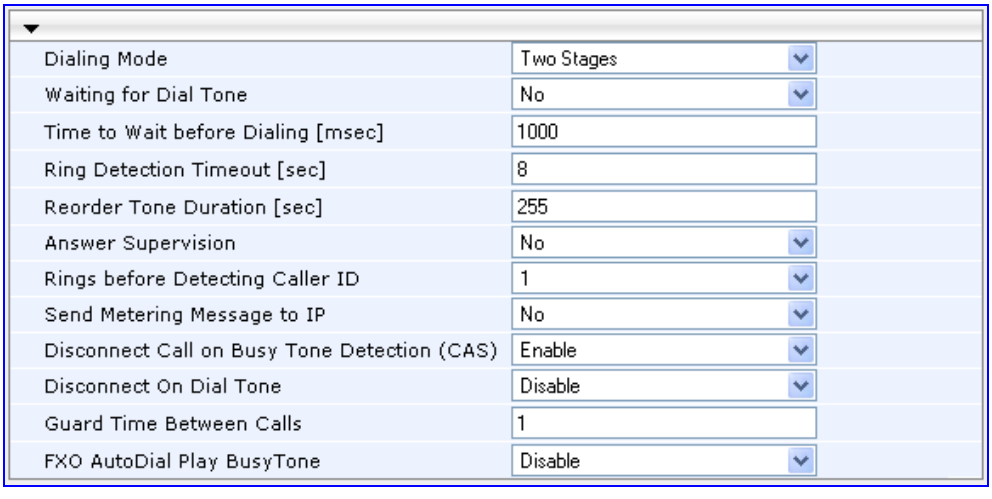
The 'FXO Settings' page allows you to configure the device's specific FXO parameters. For a description of these parameters, refer to "Configuration Parameters Reference" on page 247.

Note: The 'FXO Settings' page is available only for FXO interfaces.

➢ **To configure the FXO parameters:**

1. Open the 'FXO Settings' page (**Configuration** tab > **Advanced Applications** menu > **FXO Settings** page item).

Figure 3-112: FXO Settings Page



2. Configure the parameters as required.
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

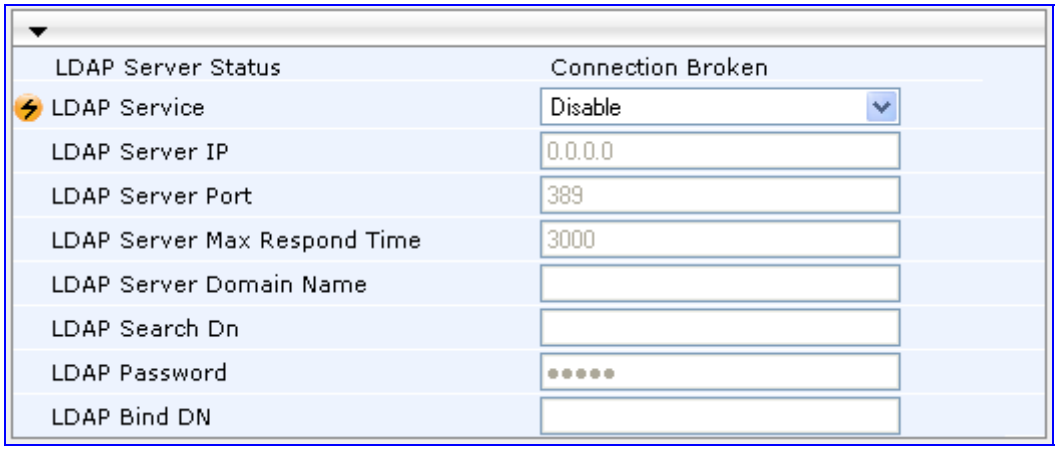
3.3.7.4 Configuring LDAP Settings

The 'LDAP Settings' page is used for configuring the Lightweight Directory Access Protocol (LDAP) parameters. For a description of these parameters, refer to "Configuration Parameters Reference" on page 247. For a detailed description of LDAP, refer to "Routing Based on LDAP Active Directory Queries" on page 537.

➢ **To configure the LDAP parameters:**

1. Open the 'LDAP Settings' page (**Configuration** tab > **Advanced Applications** menu > **LDAP Settings** page item).

Figure 3-113: LDAP Settings Page



The read-only 'LDAP Server Status' field displays one of the following possibilities:

- "Not Applicable"
- "Connection Broken"
- "Connecting"
- "Connected"

2. Configure the parameters as required.
3. Click the **Submit** button to save your changes.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

3.4 Management Tab

The **Management** tab on the Navigation bar displays menus in the Navigation tree related to device management. These menus include the following:

- Management Configuration (refer to "Management Configuration" on page 189)
- Software Update (refer to "Software Update" on page 200)

3.4.1 Management Configuration

The **Management Configuration** menu allows you to configure the device's management parameters. This menu contains the following page items:

- Management Settings (refer to "Configuring the Management Settings" on page 189)
- Regional Settings (refer to "Configuring the Regional Settings" on page 195)
- Maintenance Actions (refer to "Maintenance Actions" on page 196)

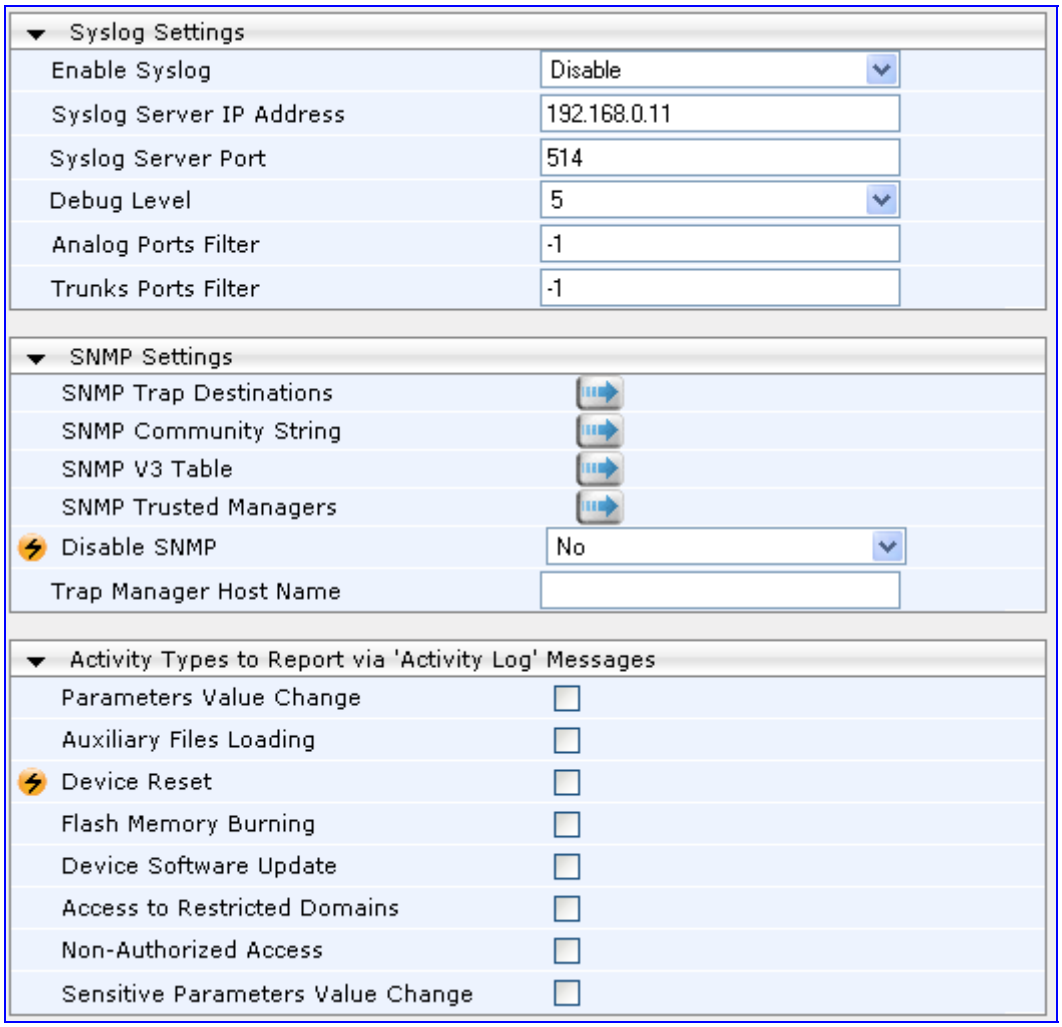
3.4.1.1 Configuring the Management Settings

The 'Management Settings' page allows you to configure the device's management parameters. For detailed description on the SNMP parameters, refer to "SNMP Parameters" on page 281.

➢ **To configure the management parameters:**

1. Open the 'Management Settings' page (**Management** tab > **Management Configuration** menu > **Management Settings** page item).

Figure 3-114: Management Settings Page



2. Configure the management parameters.
3. Configure the following SNMP tables:
   - SNMP Trap Destinations: Click the arrow button to configure the SNMP trap destinations (refer to "Configuring the SNMP Trap Destinations Table" on page 191).
   - SNMP Community String: Click the arrow button to configure the SNMP community strings (refer to "Configuring the SNMP Community Strings" on page 192).

- SNMP V3 Table: Click the arrow button to configure the SNMP V3 users (refer to "Configuring SNMP V3 Table" on page 193).
- SNMP Trusted Managers: Click the arrow button to configure the SNMP Trusted Managers (refer to "Configuring SNMP Trusted Managers" on page 194).

4. Click the **Submit** button to save your changes.
5. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

3.4.1.1.1 Configuring the SNMP Trap Destinations Table

The 'SNMP Trap Destinations' page allows you to configure up to five SNMP trap managers.

➢ **To configure the SNMP Trap Destinations table:**

1. Access the 'Management Settings' page, as described in "Configuring the Management Settings" on page 189.
2. In the 'SNMP Trap Destinations' field, click the right-pointing arrow button; the 'SNMP Trap Destinations' page appears.

Figure 3-115: SNMP Trap Destinations Page

	IP Address	Trap Port	Trap Enable
☑ SNMP Manager 1	10.8.2.28	162	Enable
☐ SNMP Manager 2	0.0.0.0	162	Enable
☐ SNMP Manager 3	0.0.0.0	162	Enable
☐ SNMP Manager 4	0.0.0.0	162	Enable
☐ SNMP Manager 5	0.0.0.0	162	Enable

3. Configure the SNMP trap managers parameters according to the table below.
4. Click the **Submit** button to save your changes.
5. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

Note: Only table row entries whose corresponding check boxes are selected are applied when clicking **Submit**; otherwise, settings revert to their defaults.

Table 3-33: SNMP Trap Destinations Parameters Description

Parameter	Description
SNMP Manager **[SNMPManagerIsUsed_x]**	Determines the validity of the parameters (IP address and port number) of the corresponding SNMP Manager used to receive SNMP traps. ▪ **[0]** (Check box cleared) = Disabled (default) ▪ **[1]** (Check box selected) = Enabled

Parameter	Description
IP Address **[SNMPManagerTableIP_x]**	IP address of the remote host used as an SNMP Manager. The device sends SNMP traps to these IP addresses. Enter the IP address in dotted-decimal notation, e.g., 108.10.1.255.
Trap Port **[SNMPManagerTrapPort_x]**	Defines the port number of the remote SNMP Manager. The device sends SNMP traps to these ports. The valid SNMP trap port range is 100 to 4000. The default port is 162.
Trap Enable **[SNMPManagerTrapSendingEnable_x]**	Activates or de-activates the sending of traps to the corresponding SNMP Manager. ▪ **[0]** Disable = Sending is disabled. ▪ **[1]** Enable = Sending is enabled (default).

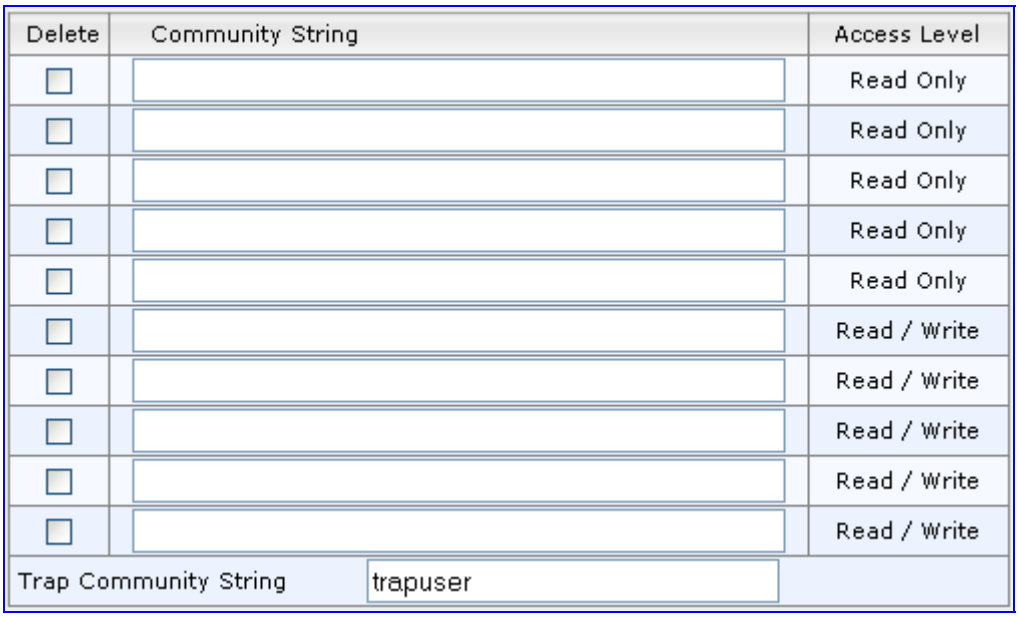
3.4.1.1.2 Configuring the SNMP Community Strings

The 'SNMP Community String' page allows you to configure up to five read-only and up to five read-write SNMP community strings, and to configure the community string that is used for sending traps. For detailed information on SNMP community strings, refer to the *Product Reference Manual*.

➢ **To configure the SNMP community strings:**

1. Access the 'Management Settings' page, as described in "Configuring the Management Settings" on page 189.
2. In the 'SNMP Community String' field, click the right-pointing arrow button; the 'SNMP Community String' page appears.

Figure 3-116: SNMP Community Strings Page



3. Configure the SNMP community strings parameters according to the table below.
4. Click the **Submit** button to save your changes.

5. To save the changes to flash memory, refer to "Saving Configuration" on page 199.

Note: To delete a community string, select the **Delete** check box corresponding to the community string that you want to delete, and then click **Submit**.

Table 3-34: SNMP Community Strings Parameters Description

Parameter	Description
Community String	▪ Read Only **[SNMPReadOnlyCommunityString_x]:** Up to five read-only community strings (up to 19 characters each). The default string is 'public'. ▪ Read / Write **[SNMPReadWriteCommunityString_x]:** Up to five read / write community strings (up to 19 characters each). The default string is 'private'.
Trap Community String **[SNMPTrapCommunityString]**	Community string used in traps (up to 19 characters). The default string is 'trapuser'.

3.4.1.1.3 Configuring SNMP V3 Users

The 'SNMP V3 Settings' page allows you to configure authentication and privacy for up to 10 SNMP v3 users.

➢ **To configure the SNMP v3 users:**

1. Access the 'Management Settings' page, as described in "Configuring the Management Settings" on page 189.
2. In the 'SNMP V3 Table' field, click the right-pointing arrow button; the 'SNMP V3 Settings' page appears.

Figure 3-117: SNMP V3 Setting Page

3. To add an SNMP v3 user, in the 'Add' field, enter the desired row index, and then click **Add**. A new row appears.
4. Configure the SNMP V3 Setting parameters according to the table below.
5. Click the **Apply** button to save your changes.
6. To save the changes, refer to "Saving Configuration" on page 199.

Notes:

- For a description of the web interface's table command buttons (e.g., **Duplicate** and **Delete**), refer to "Working with Tables" on page 39.
- You can also configure SNMP v3 users using the *ini* file table parameter SNMPUsers (refer to "SNMP Parameters" on page 281).

Table 3-35: SNMP V3 Users Parameters

Parameter	Description
Index **[SNMPUsers_Index]**	The table index. The valid range is 0 to 9.
User Name **[SNMPUsers_Username]**	Name of the SNMP v3 user. This name must be unique.
Authentication Protocol **[SNMPUsers_AuthProtocol]**	Authentication protocol of the SNMP v3 user. ▪ **[0]** None (default) ▪ **[1]** MD5 ▪ **[2]** SHA-1
Privacy Protocol **[SNMPUsers_PrivProtocol]**	Privacy protocol of the SNMP v3 user. ▪ **[0]** None (default) ▪ **[1]** DES ▪ **[2]** 3DES ▪ **[3]** AES-128 ▪ **[4]** AES-192 ▪ **[5]** AES-256
Authentication Key **[SNMPUsers_AuthKey]**	Authentication key. Keys can be entered in the form of a text password or long hex string. Keys are always persisted as long hex strings and keys are localized.
Privacy Key **[SNMPUsers_PrivKey]**	Privacy key. Keys can be entered in the form of a text password or long hex string. Keys are always persisted as long hex strings and keys are localized.
Group **[SNMPUsers_Group]**	The group with which the SNMP v3 user is associated. ▪ **[0]** Read-Only (default) ▪ **[1]** Read-Write ▪ **[2]** Trap **Note:** All groups can be used to send traps.

3.4.1.1.4 Configuring SNMP Trusted Managers

The 'SNMP Trusted Managers' page allows you to configure up to five SNMP Trusted Managers, based on IP addresses. By default, the SNMP agent accepts SNMP Get and Set requests from any IP address, as long as the correct community string is used in the request. Security can be enhanced by using Trusted Managers, which is an IP address from which the SNMP agent accepts and processes SNMP requests.

➢ **To configure the SNMP Trusted Managers:**

1. Access the 'Management Settings' page, as described in "Configuring the Management Settings" on page 189.
2. In the 'SNMP Trusted Managers' field, click the right-pointing arrow button; the 'SNMP Trusted Managers' page appears.

Figure 3-118: SNMP Trusted Managers

Delete	Trusted Managers IP Address	
☐	SNMP Trusted Manager 1	0.0.0.0
☐	SNMP Trusted Manager 2	0.0.0.0
☐	SNMP Trusted Manager 3	0.0.0.0
☐	SNMP Trusted Manager 4	0.0.0.0
☐	SNMP Trusted Manager 5	0.0.0.0

3. Select the check box corresponding to the SNMP Trusted Manager that you want to enable and for whom you want to define an IP address.
4. Define an IP address in dotted-decimal notation.
5. Click the **Submit** button to apply your changes.
6. To save the changes, refer to "Saving Configuration" on page 199.

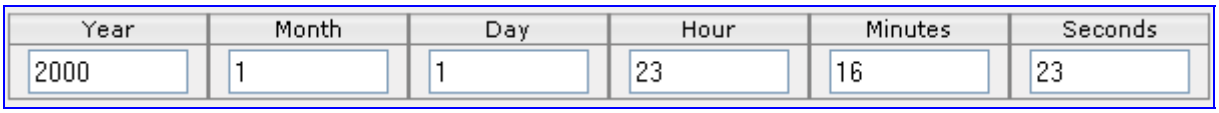
3.4.1.2 Configuring the Regional Settings

The 'Regional Settings' page allows you to define and view the device's internal date and time.

➢ **To configure the device's date and time:**

1. Open the 'Regional Settings' page (**Management** tab > **Management Configuration** menu > **Regional Settings** page item).

Figure 3-119: Regional Settings Page



2. Enter the current date and time in the geographical location in which the device is installed.
3. Click the **Submit** button; the date and time are automatically updated.

Notes:

- If the device is configured to obtain the date and time from an SNTP server (refer to "Configuring the Application Settings" on page 65), the fields on this page are read-only and cannot be modified.
- For an explanation on SNTP, refer to "Simple Network Time Protocol Support" on page 607.
- After performing a hardware reset, the date and time are returned to their defaults and therefore, should be updated.

3.4.1.3 Maintenance Actions

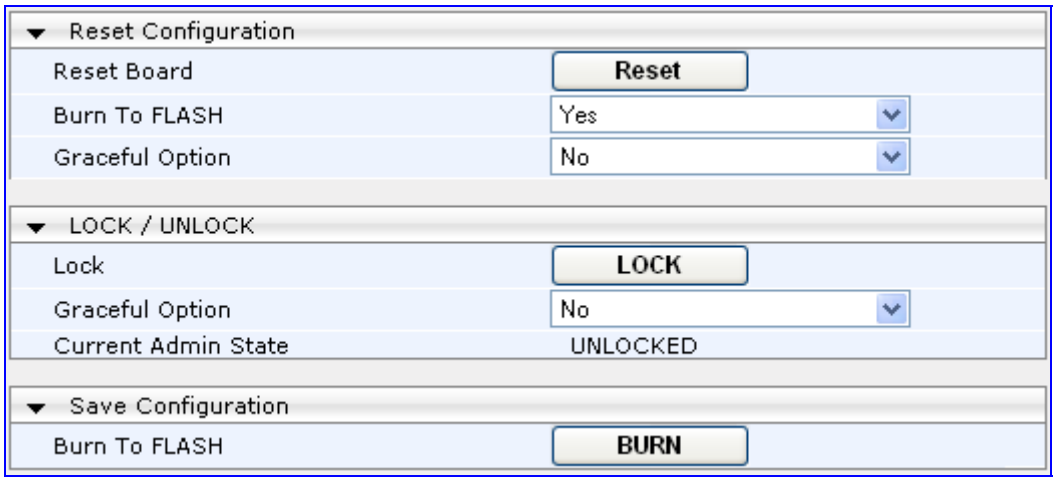
The 'Maintenance Actions' page allows you to perform the following operations:

- Reset the device (refer to "Resetting the Device" on page 197)
- Lock and unlock the device (refer to "Locking and Unlocking the Device" on page 198)
- Save the configuration to the device's flash memory (refer to "Saving Configuration" on page 199)

➢ **To access the 'Maintenance Actions' page:**

- On the Navigation bar, click the **Management** tab, and then in the Navigation tree, select the **Management Configuration** menu, and then choose the **Maintenance Actions** page item.

Figure 3-120: Maintenance Actions Page



3.4.1.3.1 Resetting the Device

The 'Maintenance Actions' page allows you to remotely reset the device. In addition, before resetting the device, you can choose the following options:

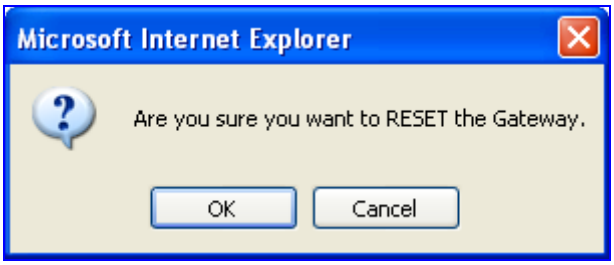
- Save the device's current configuration to the device's flash memory (non-volatile).
- Perform a graceful shutdown, i.e., device reset starts only after a user-defined time expires (i.e., timeout) or after no more active traffic exists (the earliest thereof).

➢ **To reset the device:**

1. Open the 'Maintenance Actions' page (refer to "Maintenance Actions" on page 196).
2. Under the 'Reset Configuration' group, from the 'Burn To FLASH' drop-down list, select one of the following options:
   - 'Yes': The device's current configuration is saved (*burned*) to the flash memory prior to reset (default).
   - 'No': Resets the device without saving the current configuration to flash (discards all unsaved modifications).
3. Under the 'Reset Configuration' group, from the 'Graceful Option' drop-down list, select one of the following options:
   - 'Yes': Reset starts only after the user-defined time in the 'Shutdown Timeout' field (refer to Step 4) expires or after no more active traffic exists (the earliest thereof). In addition, no new traffic is accepted.
   - 'No': Reset starts regardless of traffic, and any existing traffic is terminated at once.
4. In the 'Shutdown Timeout' field (relevant only if the 'Graceful Option' in the previous step is set to 'Yes'), enter the time after which the device resets. Note that if no traffic exists and the time has not yet expired, the device resets.

5. Click the **Reset** button; a confirmation message box appears, requesting you to confirm.

Figure 3-121: Reset Confirmation Message Box



6. Click **OK** to confirm device reset; if the parameter 'Graceful Option' is set to 'Yes' (in Step 3), the reset is delayed and a screen displaying the number of remaining calls and time is displayed. When the device begins to reset, a message appears notifying you of this.

Notes:

- Throughout the Web interface, parameters preceded by the lightning symbol are not applied on-the-fly and require that you reset the device for them to take effect.
- When you modify parameters that require a device reset, once you click the **Submit** button in the relevant page, the toolbar displays the word "Reset" (refer to "Toolbar" on page 31) to indicate that a device reset is required.

3.4.1.3.2 Locking and Unlocking the Device

The Lock and Unlock options allow you to lock the device so that it doesn't accept any new incoming calls. This is useful when, for example, you are uploading new software files to the device and you don't want any traffic to interfere with the process.

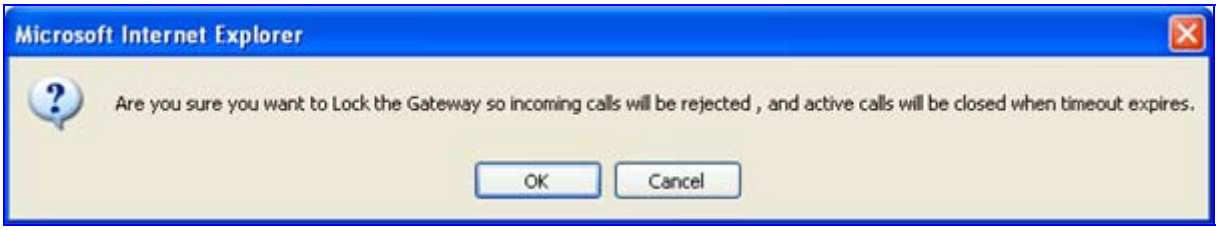
➢ **To lock the device:**

1. Open the 'Maintenance Actions' page (refer to "Maintenance Actions" on page 196).
2. Under the 'LOCK / UNLOCK' group, from the 'Graceful Option' drop-down list, select one of the following options:
   - 'Yes': The device is 'locked' only after the user-defined time in the 'Lock Timeout' field (refer to Step 3) expires or no more active traffic exists (the earliest thereof). In addition, no new traffic is accepted.
   - 'No': The device is 'locked' regardless of traffic. Any existing traffic is terminated immediately.

   Note: These options are only available if the current status of the device is in the Unlock state.
3. In the 'Lock Timeout' field (relevant only if the parameter 'Graceful Option' in the previous step is set to 'Yes'), enter the time (in seconds) after which the device locks. Note that if no traffic exists and the time has not yet expired, the device locks.

4. Click the **LOCK** button; a confirmation message box appears requesting you to confirm device Lock.

Figure 3-122: Device Lock Confirmation Message Box


Microsoft Internet Explorer
Are you sure you want to Lock the Gateway so incoming calls will be rejected , and active calls will be closed when timeout expires.
OK
Cancel

5. Click **OK** to confirm device Lock; if 'Graceful Option' is set to 'Yes', the lock is delayed and a screen displaying the number of remaining calls and time is displayed. Otherwise, the lock process begins immediately. The 'Current Admin State' field displays the current state: LOCKED or UNLOCKED.

➢ **To unlock the device:**

1. Open the 'Maintenance Actions' page (refer to "Maintenance Actions" on page 196).
2. Under the 'LOCK / UNLOCK' group, click the **UNLOCK** button. Unlock starts immediately and the device accepts new incoming calls.

3.4.1.3.3 Saving Configuration

The 'Maintenance Actions' page allows you to save (*burn*) the current parameter configuration (including loaded auxiliary files) to the device's *non-volatile* memory (i.e., flash). The parameter modifications that you make throughout the Web interface's pages are temporarily saved (to the *volatile* memory - RAM) when you click the **Submit** button on these pages. Parameter settings that are only saved to the device's RAM revert to their previous settings after a hardware/software reset (or power failure). Therefore, to ensure that your configuration changes are retained, you must save them to the device's flash memory using the burn option described below.

➢ **To save the changes to the non-volatile flash memory :**

1. Open the 'Maintenance Actions' page (refer to "Maintenance Actions" on page 196).
2. Under the 'Save Configuration' group, click the **BURN** button; a confirmation message appears when the configuration successfully saves.

Notes:

- Saving configuration to the *non-volatile* memory may disrupt current traffic on the device. To avoid this, disable all new traffic before saving, by performing a graceful lock (refer to "Locking and Unlocking the Device" on page 198).
- Throughout the Web interface, parameters preceded by the lightning symbol are not applied on-the-fly and require that you reset the device for them to take effect (refer to "Resetting the Device" on page 197).

3.4.2 Software Update

The **Software Update** menu allows you to upgrade the device's software by loading a new *cmp* file (compressed firmware) along with the *ini* file and a suite of auxiliary files. This menu includes the following page items:

- Load Auxiliary Files (refer to "Loading Auxiliary Files" on page 200)
- Software Upgrade Key (refer to "Loading a Software Upgrade Key" on page 202)
- Software Upgrade Wizard (refer to "Software Upgrade Wizard" on page 205)
- Configuration File (refer to "Backing Up and Restoring Configuration" on page 208)

3.4.2.1 Loading Auxiliary Files

The 'Load Auxiliary Files' page allows you to load various auxiliary files to the device. These auxiliary files are briefly described in the table below:

Table 3-36: Auxiliary Files Descriptions

File Type	Description
ini	Provisions the device's parameters. The Web interface enables practically full device provisioning, but customers may occasionally require new feature configuration parameters in which case this file is loaded. **Note:** Loading this file only provisions those parameters that are included in the *ini* file. Parameters that are not specified in the *ini* file are reset to factory default values.
CAS	Up to eight different CAS files containing specific CAS protocol definitions for digital modules. These files are provided to support various types of CAS signaling.
Voice Prompts	The voice announcement file contains a set of Voice Prompts (VP) that are played by the device during operation. **Note:** This file is applicable only to Mediant 1000.
Call Progress Tones	This is a region-specific, telephone exchange-dependent file that contains the Call Progress Tones (CPT) levels and frequencies that the device uses. The default CPT file is U.S.A.
Prerecorded Tones	The *dat* PRT file enhances the device's capabilities of playing a wide range of telephone exchange tones that cannot be defined in the Call Progress Tones file.
Dial Plan	Dial plan file.
User Info	The User Information file maps PBX extensions to IP numbers. This file can be used to represent PBX extensions as IP phones in the global 'IP world'.

Notes:

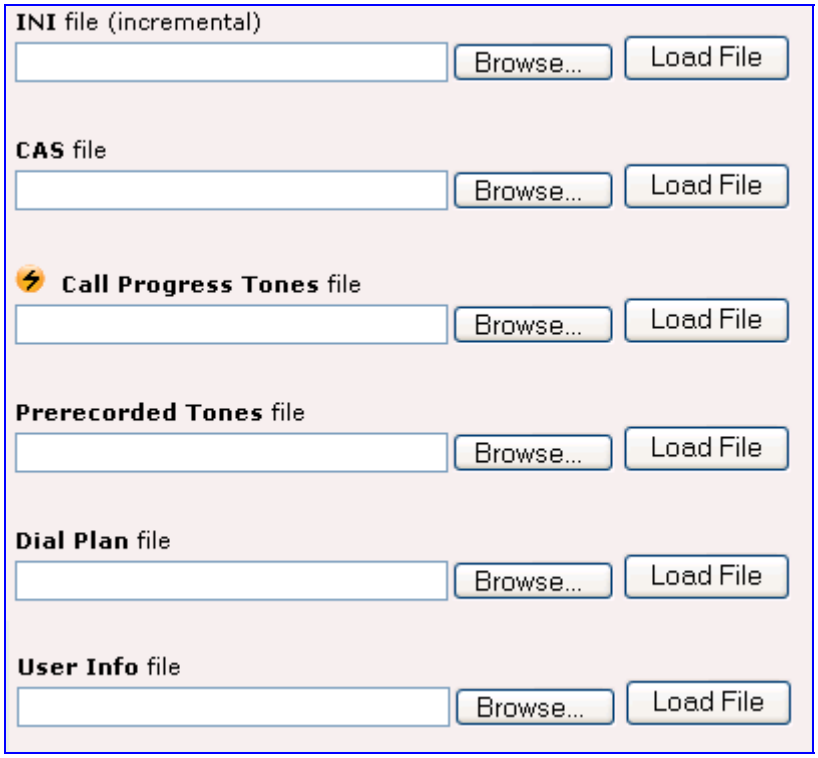
- You can schedule automatic loading of updated auxiliary files using HTTP/HTTPS, FTP, or NFS (refer to the *Product Reference Manual*).
- For a detailed description on auxiliary files, refer to "Auxiliary Configuration Files" on page 473.
- When loading an *ini* file, the current settings of parameters that are excluded from the loaded *ini* file are retained (*incremental*).
- Saving an auxiliary file to flash memory may disrupt traffic on the device. To avoid this, disable all traffic on the device, by performing a graceful lock (refer to "Locking and Unlocking the Device" on page 198).
- For deleting auxiliary files, refer to "Viewing Device Information" on page 211.

The auxiliary files can be loaded to the device using the Web interface's 'Load Auxiliary Files' page, as described in the procedure below.

➢ **To load an auxiliary file to the device using the Web interface:**

1. Open the 'Load Auxiliary Files' page (**Management** tab > **Software Update** menu > **Load Auxiliary Files** page item).

Figure 3-123: Load Auxiliary Files Page



2. Click the **Browse** button corresponding to the file type that you want to load, navigate to the folder in which the file is located, and then click **Open**; the name and path of the file appear in the field next to the **Browse** button.
3. Click the **Load File** button corresponding to the file you want to load.

4. Repeat steps 2 through 3 for each file you want to load.
5. To save the loaded auxiliary files to flash memory, refer to "Saving Configuration" on page 199.
6. To reset the device (if you have loaded a Call Progress Tones file), refer to "Resetting the Device" on page 197.

You can also load the auxiliary files using the *ini* file (loaded to the device using BootP). Each auxiliary file has a specific *ini* file parameter that specifies the name of the auxiliary file that you want to load to the device. For a description of these *ini* file parameters, refer to Configuration Files Parameters on page 467.

➢ **To load the auxiliary files using an *ini* file:**

1. In the *ini* file, define the auxiliary files to be loaded to the device. You can also define in the *ini* file whether the loaded files must be stored in the non-volatile memory so that the TFTP process is not required every time the device boots up.
2. Save the auxiliary files and the *ini* file in the same directory on your local PC.
3. Invoke a BootP/TFTP session; the *ini* and associated auxiliary files are loaded to the device.

3.4.2.2 Loading a Software Upgrade Key

The 'Software Upgrade Key Status' page allows you to load a new Software Upgrade Key to the device. The device is supplied with a Software Upgrade Key, which determines the device's supported features, capabilities, and available resources. You can upgrade or change your device's supported items by purchasing a new Software Upgrade Key to match your requirements.

The Software Upgrade Key is provided in string format, in a text-based file. When you load a Software Upgrade Key, it is loaded to the device's non-volatile flash memory, and overwrites the previously installed key.

You can load a Software Upgrade Key using one of the following management tools:

- Web interface
- BootP/TFTP configuration utility (refer to Loading via BootP/TFTP on page 204)
- AudioCodes' EMS (refer to *EMS User's Manual* or *EMS Product Description*)

Warning: Do not modify the contents of the Software Upgrade Key file.

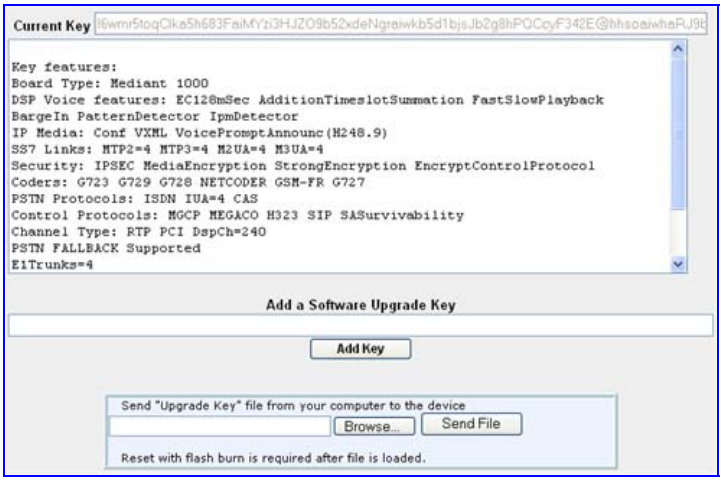
Note: The Software Upgrade Key is an encrypted key.

The procedure below describes how to load a Software Upgrade Key to the device using the Web interface.

➢ To load a Software Upgrade Key:

1. Open the 'Software Upgrade Key Status' page (**Management** tab > **Software Update** menu > **Software Upgrade Key** page item).

Figure 3-124: Software Upgrade Key Status Page


Current Key
Key features:
Board Type: Mediant 1000
DSP Voice features: EC128mSec AdditionTimeslotSummation FastSlowPlayback
BargeIn PatternDetector IpmDetector
IP Media: Conf VXML VoicePromptAnnounc(H248.9)
SS7 Links: MTP2=4 MTP3=4 M2UA=4 M3UA=4
Security: IPSEC MediaEncryption StrongEncryption EncryptControlProtocol
Coders: G723 G729 G728 NETCODER GSM-FR G727
PSTN Protocols: ISDN IUA=4 CAS
Control Protocols: MGCP MEGACO H323 SIP SASurvivability
Channel Type: RTP PCI DspCh=240
PSTN FALLBACK Supported
E1Trunks=4
Add a Software Upgrade Key
Add Key
Send "Upgrade Key" file from your computer to the device
Browse...
Send File
Reset with flash burn is required after file is loaded.

2. Backup your current Software Upgrade Key as a precaution so that you can re-load this backup key to restore the device's original capabilities if the new key doesn't comply with your requirements:
   a. In the 'Current Key' field, copy the string of text and paste it in any standard text file.
   b. Save the text file to a folder on your PC with a name of your choosing.
3. Open the new Software Upgrade Key file and ensure that the first line displays '**[LicenseKeys]**' and that it contains one or more lines in the following format: S/N<serial number> = <long Software Upgrade Key>

   For example: S/N370604 = jCx6r5tovCIKaBBbhPtT53Yj...

   One S/N must match the serial number of your device. The device's serial number can be viewed in the 'Device Information' page (refer to "Viewing Device Information" on page 211).
4. Follow one of the following procedures, depending on whether you are loading a single or multiple key S/N lines:
   - **Single key S/N line:**
     a. Open the Software Upgrade Key text file (using, for example, Microsoft® Notepad).
     b. Select and copy the key string of the device's S/N and paste it into the field 'Add a Software Upgrade Key'.
     c. Click the **Add Key** button.

- **Multiple S/N lines (as shown below):**

Figure 3-125: Software Upgrade Key with Multiple S/N Lines

a. in the 'Send Upgrade Key file' field, click the **Browse** button and navigate to the folder in which the Software Upgrade Key text file is located on your PC.

b. Click the **Send File** button; the new key is loaded to the device and validated. If the key is valid, it is burned to memory and displayed in the 'Current Key' field.

5. Verify that the Software Upgrade Key file was successfully loaded to the device, by using one of the following methods:

- In the 'Key features' group, ensure that the features and capabilities activated by the installed string match those that were ordered.
- Access the Syslog server (refer to the *Product Reference Manual*) and ensure that the following message appears in the Syslog server:
"S/N___ Key Was Updated. The Board Needs to be Reloaded with *ini* file\n".

6. Reset the device; the new capabilities and resources are active.

Note: If the Syslog server indicates that the Software Upgrade Key file was unsuccessfully loaded (i.e., the 'SN_' line is blank), perform the following preliminary troubleshooting procedures:

1. Open the Software Upgrade Key file and check that the S/N line appears. If it does not appear, contact AudioCodes.
2. Verify that you've loaded the correct file. Open the file and ensure that the first line displays **[LicenseKeys].**
3. Verify that the contents of the file has not been altered in any way.

3.4.2.2.1 Loading via BootP/TFTP

The procedure below describes how to load a Software Upgrade Key to the device using AudioCodes' BootP/TFTP Server utility (for a detailed description on the BootP utility, refer to the *Product Reference Manual*).

➢ **To load a Software Upgrade Key file using BootP/TFTP:**

1. Place the Software Upgrade Key file (typically, a *.txt file) in the same folder in which the device's *cmp* file is located.
2. Start the BootP/TFTP Server utility.
3. From the **Services** menu, choose **Clients**; the 'Client Configuration' screen is displayed.

4. From the 'INI File' drop-down list, select the Software Upgrade Key file. Note that the device's *cmp* file must be specified in the 'Boot File' field.
5. Configure the initial BootP/TFTP parameters as required, and then click **OK**.
6. Reset the device; the *cmp* and Software Upgrade Key files are loaded to the device.

Note: To load the Software Upgrade Key using BootP/TFTP, the extension name of the key file must be *.*ini*.

3.4.2.3 Software Upgrade Wizard

The Software Upgrade Wizard allows you to upgrade the device's firmware (cmp file) as well as load an *ini* file and/or auxiliary files (e.g., Call Progress Tones). However, it is mandatory, when using the wizard to first load a *cmp* file to the device. You can then choose to also load an *ini* file and/or auxiliary files, but this cannot be pursued without first loading a *cmp* file. For the *ini* and each auxiliary file type, you can choose to load a new file, or not load a file but use the existing file (i.e., maintain existing configuration) running on the device.

The Software Upgrade Wizard allows you to load the following files:

- **cmp:** (Mandatory) compressed firmware file
- **Optional files:**
  - ***ini*:** configuration file
  - **Auxiliary files:** CPT (Call Progress Tone), VP (Voice Prompts), PRT (Prerecorded Tones), CAS, and USERINF (User Information)

Warnings:

- To preserve all configuration settings, before upgrading the device to a new major software version (e.g., from version 5.8 to 6.0), save a copy of the device's configuration settings (i.e., *ini* file) to your PC and ensure that you have all the original auxiliary files currently used by the device. After you have upgraded the device, restore your configuration settings by uploading these files to the device. For saving and restoring configuration, refer to "Backing Up and Restoring Configuration" on page 208.
- The Software Upgrade Wizard requires the device to be reset at the end of the process, which may disrupt traffic. To avoid this, disable all traffic on the device before initiating the wizard by performing a graceful lock (refer to Saving and Resetting the Device).

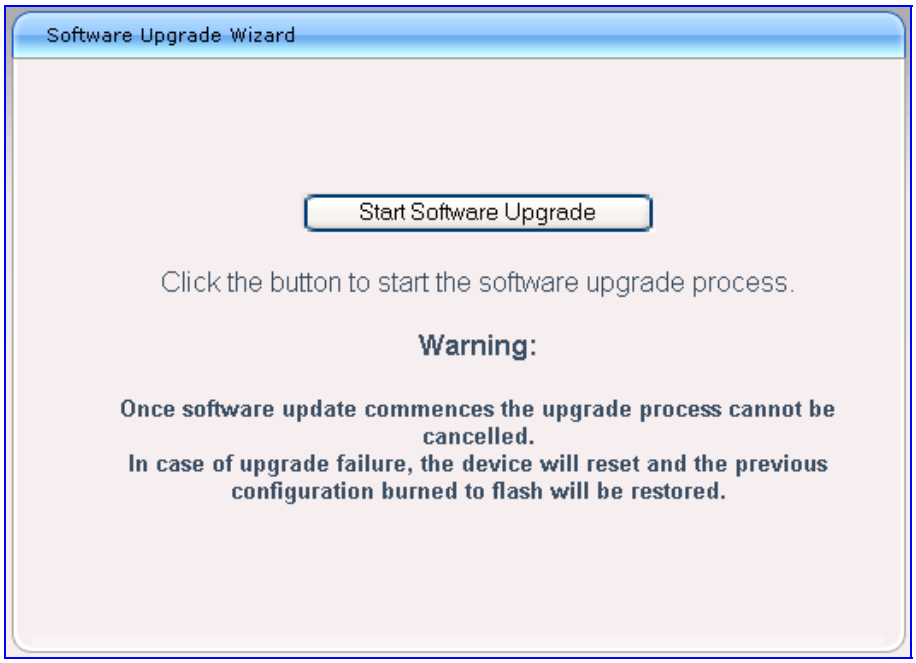
Notes:

- Before you can load an *ini* or any auxiliary file, you must first load a *cmp* file.
- When you activate the wizard, the rest of the Web interface is unavailable. After the files are successfully loaded, access to the full Web interface is restored.
- If you upgraded your cmp and the "SW version mismatch" message appears in the Syslog or Web interface, you know that your Software Upgrade Key does not support the new cmp version. Contact AudioCodes support for assistance.
- You can schedule automatic loading of these files using HTTP/HTTPS, FTP, or NFS (refer to the *Product Reference Manual*).

➢ **To load files using the Software Upgrade Wizard:**

1. Stop all traffic on the device using the Graceful Lock feature (refer to the warning bulletin above).
2. Open the 'Software Upgrade Wizard' (**Management** tab > **Software Update** menu > **Software Upgrade Wizard**); the 'Software Upgrade Wizard' page appears.

Figure 3-126: Start Software Upgrade Wizard Screen


Software Upgrade Wizard
Start Software Upgrade
Click the button to start the software upgrade process.
Warning:
Once software update commences the upgrade process cannot be cancelled.
In case of upgrade failure, the device will reset and the previous configuration burned to flash will be restored.

3. Click the **Start Software Upgrade** button; the 'Load a CMP file' Wizard page appears.

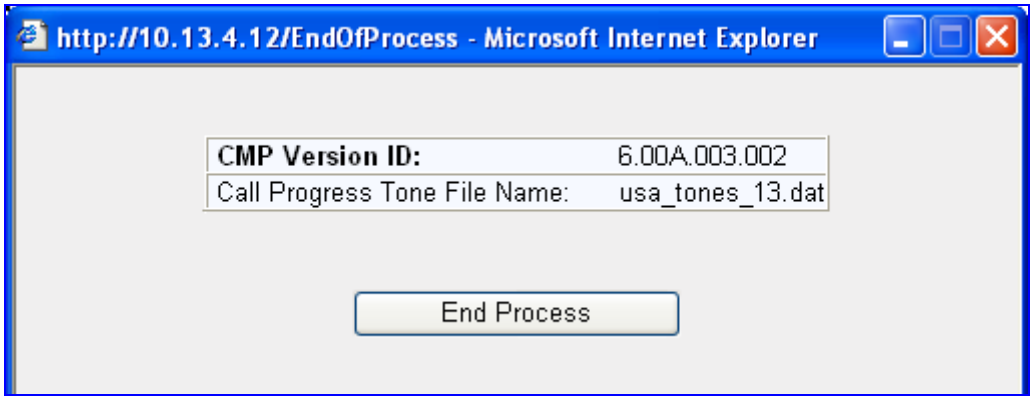
Note: At this stage, you can quit the Software Update Wizard, by clicking **Cancel**, without requiring a device reset. However, once you start uploading a cmp file, the process must be completed with a device reset.

4. Click the **Browse** button, navigate to the *cmp* file, and then click **Send File**; the *cmp* file is loaded to the device and you're notified as to a successful loading.

5. Click one of the following buttons:
   - **Reset**; the device resets with the newly loaded cmp, utilizing the existing configuration and auxiliary files.
   - **Next**; the 'Load an *ini* File' wizard page opens.

   Note that as you progress by clicking **Next**, the relevant file name corresponding to the applicable Wizard page is highlighted in the file list on the left.
6. In the 'Load an *ini* File' page, you can now choose to either:
   - Click **Browse**, navigate to the *ini* file, and then click **Send File**; the *ini* file is loaded to the device and you're notified as to a successful loading.
   - Use the *ini* file currently used by the device, by not selecting an *ini* file and by ensuring that the 'Use existing configuration' check box is marked (default).
   - Return the device's configuration settings to factory defaults, by not selecting an *ini* file and by clearing the 'Use existing configuration' check box.
7. You can now choose to either:
   - Click **Reset**; the device resets, utilizing the new *cmp* and *ini* file you loaded up to now as well as utilizing the other auxiliary files.
   - Click **Back**; the 'Load a *cmp* file' page is opened again.
   - Click **Next**; the next page opens for loading the next consecutive auxiliary file listed in the Wizard.
8. For loading the auxiliary files, follow the same procedure as for loading the *ini* file (Step 6).
9. In the 'FINISH' page, complete the upgrade process by clicking **Reset;** the device 'burns' the newly loaded files to flash memory and then resets the device. After the device resets, the 'End Process' screen appears displaying the burned configuration files (refer to the figure below).

Figure 3-127: End Process Wizard Page


http://10.13.4.12/EndOfProcess - Microsoft Internet Explorer
CMP Version ID: 6.00A.003.002
Call Progress Tone File Name: usa_tones_13.dat
End Process

10. Click **End Process** to close the wizard; the 'Enter Network Password' dialog box appears.
11. Enter your login user name and password, and then click **OK**; a message box appears informing you of the new cmp file.
12. Click **OK**; the Web interface becomes active, reflecting the upgraded device.

3.4.2.4 Backing Up and Restoring Configuration

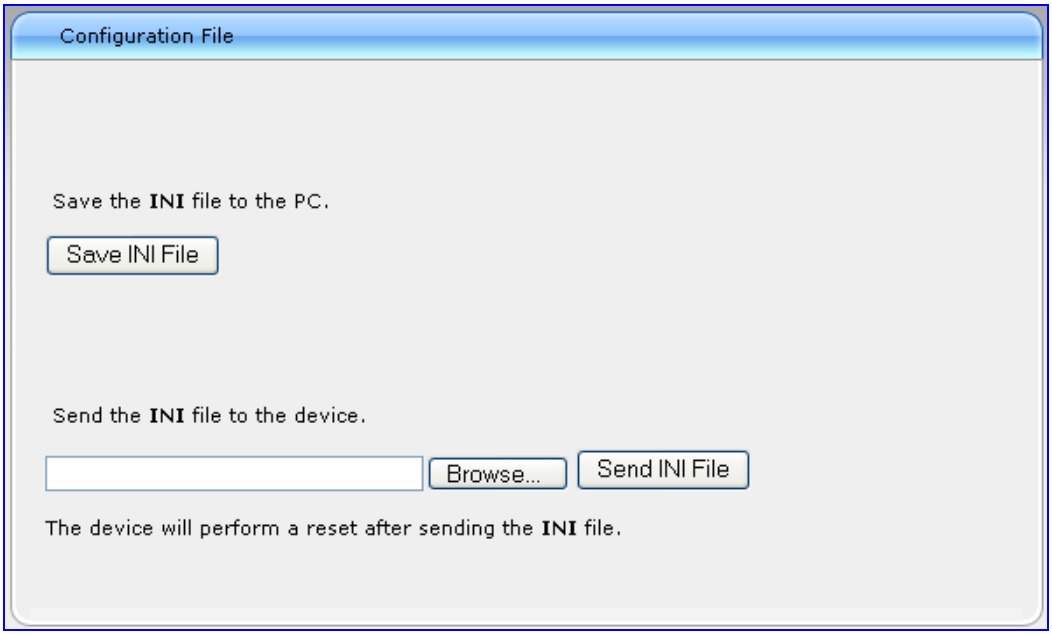
You can save a copy/backup of the device's current configuration settings as an *ini* file to a folder on your PC, using the 'Configuration File' page. The saved *ini* file includes only parameters that were modified and parameters with other than default values. The 'Configuration File' page also allows you to load an *ini* file to the device. If the device has "lost" its configuration, you can restore the device's configuration by loading the previously saved *ini* file or by simply loading a newly created *ini* file.

Note: When loading an *ini* file using this Web page, parameters not included in the *ini* file are reset to default settings.

➢ **To save and restore the *ini* file:**

1. Open the 'Configuration File' page (**Management** tab > **Software Update** menu > **Configuration File**).

Figure 3-128: Configuration File Page


Configuration File
Save the INI file to the PC.
Save INI File
Send the INI file to the device.
Browse...
Send INI File
The device will perform a reset after sending the INI file.

2. Click the **Save INI File** button; the 'File Download' dialog box appears.
3. Click the **Save** button, navigate to the folder in which you want to save the *ini* file on your PC, and then click **Save**; the device copies the *ini* file to the selected folder.

➢ **To load (or restore) the *ini* file:**

1. Click the **Browse** button, navigate to the folder in which the *ini* file is located, select the file, and then click **Open**; the name and path of the file appear in the field beside the **Browse** button.
2. Click the **Load INI File** button, and then at the prompt, click **OK**; the device uploads the *ini* file and then resets (from the *cmp* version stored on the flash memory). Once complete, the 'Enter Network Password' dialog box appears, requesting you to enter your user name and password.

3.5 Status & Diagnostics Tab

The **Status & Diagnostics** tab on the Navigation bar displays menus in the Navigation tree related to device operating status and diagnostics. These menus include the following:

- Status & Diagnostics (refer to "Status & Diagnostics" on page 209)
- Gateway Statistics (refer to "Gateway Statistics" on page 214)

3.5.1 Status & Diagnostics

The **Status & Diagnostics** menu is used to view and monitor the device's channels, Syslog messages, hardware and software product information, and to assess the device's statistics and IP connectivity information. This menu includes the following page items:

- Message Log (refer to Viewing the Device's Syslog Messages on page 209)
- Ethernet Port Information (refer to "Viewing Ethernet Port Information" on page 210)
- IP Interface Status (refer to "Viewing Active IP Interfaces" on page 211)
- Device Information (refer to "Viewing Device Information" on page 211)
- Performance Statistics (refer to "Viewing Performance Statistics" on page 212)
- Active Alarms (refer to "Viewing Active Alarms" on page 213)

3.5.1.1 Viewing the Device's Syslog Messages

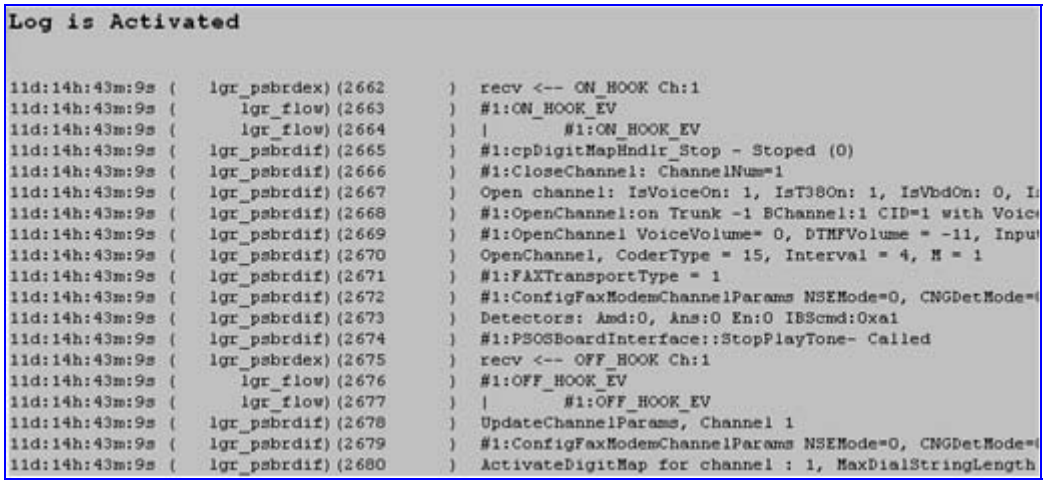
The 'Message Log' page displays Syslog debug messages sent by the device. You can select the Syslog messages in this page, and then copy and paste them into a text editor such as Notepad. This text file (*txt*) can then be sent to AudioCodes Technical Support for diagnosis and troubleshooting.

Note: It's not recommended to keep a Message Log session open for a prolonged period. This may cause the device to overload. For prolonged (and detailed) debugging, use an external Syslog server (refer to the *Product Reference Manual*).

➢ **To activate the Message Log:**

1. Set the parameter 'Debug Level' (GwDebugLevel) to 7 (refer "Configuring Advanced Parameter" on page 135). This parameter determines the Syslog logging level in the range 0 to 6, where 7 is the highest level.
2. Open the 'Message Log' page (**Status & Diagnostics** tab > **Status & Diagnostics** menu > **Message Log** page item); the 'Message Log' page is displayed and the log is activated.

Figure 3-129: Message Log Screen


Log is Activated
11d:14h:43m:9s (lgr_psbrdex)(2662) recv <-- ON_HOOK Ch:1
11d:14h:43m:9s (lgr_flow)(2663) #1:ON_HOOK_EV
11d:14h:43m:9s (lgr_flow)(2664) | #1:ON_HOOK_EV
11d:14h:43m:9s (lgr_psbrdif)(2665) #1:cpDigitMapHndlr_Stop - Stoped (0)
11d:14h:43m:9s (lgr_psbrdif)(2666) #1:CloseChannel: ChannelNum=1
11d:14h:43m:9s (lgr_psbrdif)(2667) Open channel: IsVoiceOn: 1, IsT38On: 1, IsVbdOn: 0, I
11d:14h:43m:9s (lgr_psbrdif)(2668) #1:OpenChannel:on Trunk -1 BChannel:1 CID=1 with Voic
11d:14h:43m:9s (lgr_psbrdif)(2669) #1:OpenChannel VoiceVolume= 0, DTMFVolume = -11, Inpu
11d:14h:43m:9s (lgr_psbrdif)(2670) OpenChannel, CoderType = 15, Interval = 4, M = 1
11d:14h:43m:9s (lgr_psbrdif)(2671) #1:FAXTransportType = 1
11d:14h:43m:9s (lgr_psbrdif)(2672) #1:ConfigFaxModemChannelParams NSEMode=0, CNGDetMode=
11d:14h:43m:9s (lgr_psbrdif)(2673) Detectors: Amd:0, Ans:0 En:0 IBScmd:0xa1
11d:14h:43m:9s (lgr_psbrdif)(2674) #1:PSOSBoardInterface::StopPlayTone- Called
11d:14h:43m:9s (lgr_psbrdex)(2675) recv <-- OFF_HOOK Ch:1
11d:14h:43m:9s (lgr_flow)(2676) #1:OFF_HOOK_EV
11d:14h:43m:9s (lgr_flow)(2677) | #1:OFF_HOOK_EV
11d:14h:43m:9s (lgr_psbrdif)(2678) UpdateChannelParams, Channel 1
11d:14h:43m:9s (lgr_psbrdif)(2679) #1:ConfigFaxModemChannelParams NSEMode=0, CNGDetMode=
11d:14h:43m:9s (lgr_psbrdif)(2680) ActivateDigitMap for channel : 1, MaxDialStringLength

The displayed logged messages are color coded as follows:

- Yellow - fatal error message
- Blue - recoverable error message (i.e., non-fatal error)
- Black - notice message

3. To clear the page of Syslog messages, access the 'Message Log' page again (see Step 2); the page is cleared and new messages begin appearing.

➢ **To stop the Message Log:**

■ Close the 'Message Log' page by accessing any another page in the Web interface.

3.5.1.2 Viewing Ethernet Port Information

The 'Ethernet Port Information' page displays read-only information on the Ethernet connection used by the device. This includes indicating the active port, duplex mode, and speed. You can also access this page from the 'Home' page (refer to "Using the Home Page" on page 54).

For detailed information on the Ethernet redundancy scheme, refer to Ethernet Interface Redundancy on page 603. For detailed information on the Ethernet interface configuration, refer to "Ethernet Interface Configuration" on page 603.

➢ **To view Ethernet port information:**

- Open the 'Ethernet Port Information' page (**Status & Diagnostics** tab > **Status & Diagnostics** menu > **Ethernet Port Information** page item).

Figure 3-130: Ethernet Port Information Screen

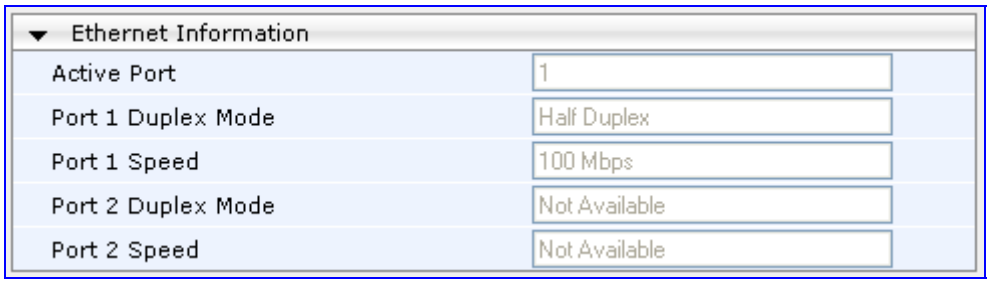


Table 3-37: Ethernet Port Information Parameters

Parameter	Description
Active Port	Displays the active Ethernet port (1 or 2).
Port Duplex Mode	Displays the Duplex mode of the Ethernet port.
Port Speed	Displays the speed (in Mbps) of the Ethernet port.

3.5.1.3 Viewing Active IP Interfaces

The 'IP Interface Status' page displays the device's active IP interfaces, which are configured in the 'Multiple Interface Table' page (refer to "Configuring the Multiple Interface Table" on page 61).

➢ **To view the 'Active IP Interfaces' page:**

- Open the 'IP Interface Status' page (**Status & Diagnostics** tab > **Status & Diagnostics** menu > **IP Interface Status** page item).

Figure 3-131: IP Interface Status Page

Index	Application Type	Address Type	Interface Mode	IP Address	Prefix Length	Gateway	VLAN ID	Interface Name
NA	O+M+C	IPv4	IPv4 Manual	10.8.7.31	16	10.8.0.1	0	All

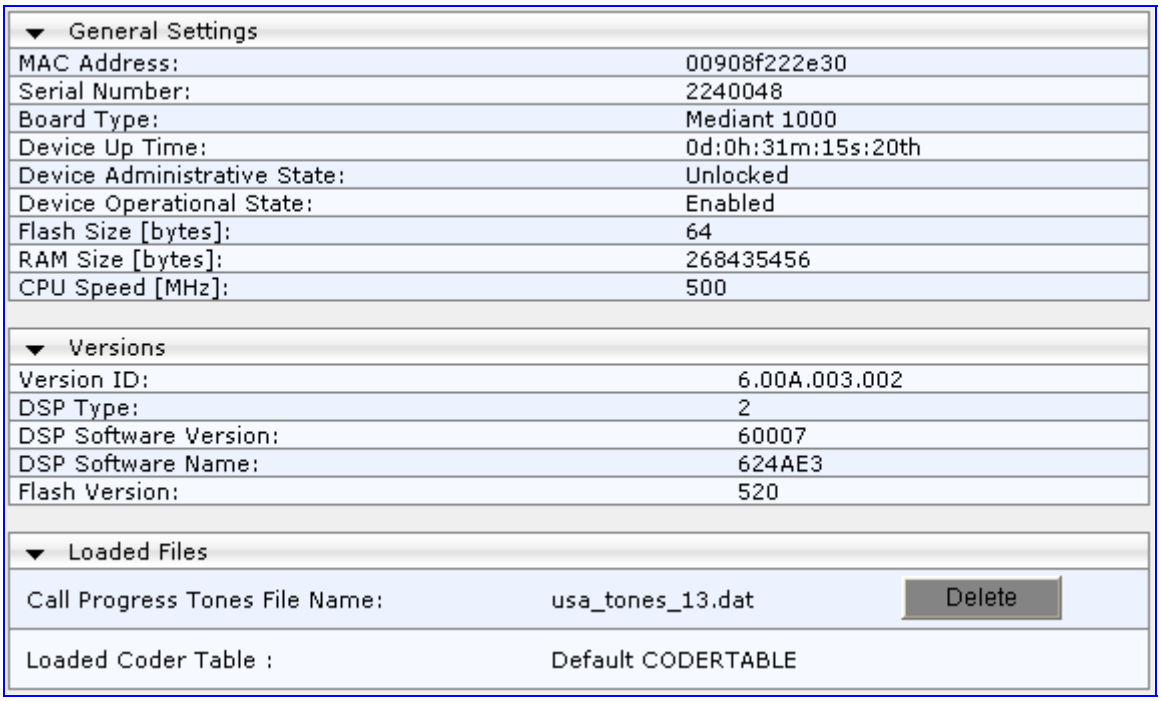
3.5.1.4 Viewing Device Information

The 'Device Information' page displays the device's specific hardware and software product information. This information can help you expedite troubleshooting. Capture the page and e-mail it to AudioCodes Technical Support personnel to ensure quick diagnosis and effective corrective action. This page also displays any loaded files used by the device (stored in the RAM) and allows you to remove them.

➢ **To access the 'Device Information' page:**

- Open the 'Device Information' page (**Status & Diagnostics** tab > **Status & Diagnostics** menu > **Device Information** page item).

Figure 3-132: Device Information Page



➢ **To delete a loaded file:**

- Click the **Delete** button corresponding to the file that you want to delete. Deleting a file takes effect only after device reset (refer to "Resetting the Device" on page 197).

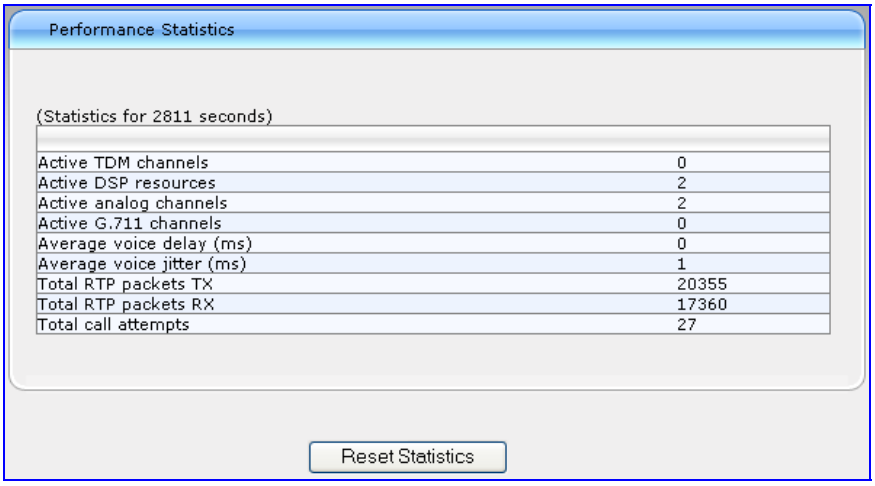
3.5.1.5 Viewing Performance Statistics

The 'Performance Statistics' page provides read-only, device performance statistics. This page is refreshed with new statistics every 60 seconds. The duration that the current statistics has been collected, is displayed above the statistics table.

➢ **To view performance statistics:**

- Open the 'Performance Statistics' page (**Status & Diagnostics** tab > **Status & Diagnostics** menu > **Performance Statistics** page item).

Figure 3-133: Performance Statistics Page


Performance Statistics
(Statistics for 2811 seconds)
Active TDM channels 0
Active DSP resources 2
Active analog channels 2
Active G.711 channels 0
Average voice delay (ms) 0
Average voice jitter (ms) 1
Total RTP packets TX 20355
Total RTP packets RX 17360
Total call attempts 27
Reset Statistics

➢ **To reset the performance statistics to zero:**

- Click the **Reset Statistics** button.

3.5.1.6 Viewing Active Alarms

The 'Active Alarms' page displays a list of currently active alarms. You can also access this page from the 'Home' page (refer to "Using the Home Page" on page 54).

➢ **To view the list of alarms:**

- Open the 'Active Alarms' page (**Status & Diagnostics** tab > **Status & Diagnostics** menu > **Active Alarms** page item).

For each alarm, the following information is provided:

- **Severity:** severity level of the alarm:
  - Critical - alarm displayed in red
  - Major - alarm displayed in orange
  - Minor - alarm displayed in yellow
- **Source:** unit from which the alarm was raised
- **Description:** brief explanation of the alarm
- **Date:** date and time that the alarm was generated

You can view the next 30 alarms (if exist), by pressing the F5 key.

3.5.2 Gateway Statistics

The **Gateway Statistics** menu allows you to monitor real-time activity such as IP connectivity information, call details and call statistics, including the number of call attempts, failed calls, fax calls, etc. This menu includes the following page items:

- IP to Tel Calls Count (refer to "Viewing Call Counters" on page 214)
- Tel to IP Calls Count (refer to "Viewing Call Counters" on page 214)
- SAS Registered Users (refer to “Viewing SAS Registered Users” on page 216)
- Call Routing Status (refer to "Viewing Call Routing Status" on page 217)
- Registration Status (refer to “Viewing Registration Status” on page 217)
- IP Connectivity (refer to "Viewing IP Connectivity" on page 219)

Note: The Web pages pertaining to the **Gateway Statistics** menu do not refresh automatically. To view updated information, close the relevant page and then re-access it.

3.5.2.1 Viewing Call Counters

The 'IP to Tel Calls Count' and 'Tel to IP Calls Count' pages provide you with statistical information on incoming (IP-to-Tel) and outgoing (Tel-to-IP) calls. The statistical information is updated according to the release reason that is received after a call is terminated (during the same time as the end-of-call Call Detail Record or CDR message is sent). The release reason can be viewed in the 'Termination Reason' field in the CDR message.

You can reset the statistical data displayed on the page (i.e., refresh the display), by clicking the **Reset Counters** button located on the page.

➢ **To view the IP-to-Tel and Tel-to-IP Call Counters pages:**

- Open the Call Counters page that you want to view (**Status & Diagnostics** tab > **Gateway Statistics** menu > **IP to Tel Calls Count** or **Tel to IP Calls Count** page item); the figure below shows the 'IP to Tel Calls Count' page.

Figure 3-134: Calls Count Page

Number of Attempted Calls	19
Number of Established Calls	14
Percentage of Successful Calls(ASR)	73.684211
Number of Calls Terminated due to a Busy Line	2
Number of Calls Terminated due to No Answer	0
Number of Calls Terminated due to Forward	0
Number of Failed Calls due to No Route	0
Number of Failed Calls due to No Matched Capabilities	0
Number of Failed Calls due to No Resources	0
Number of Failed Calls due to Other Failures	0
Average Call Duration(ACD)[sec]	25
Attempted Fax Calls Counter	0
Successful Fax Calls Counter	0

Table 3-38: Call Counters Description

Counter	Description
Number of Attempted Calls	Indicates the number of attempted calls. It is composed of established and failed calls. The number of established calls is represented by the 'Number of Established Calls' counter. The number of failed calls is represented by the failed-call counters. Only one of the established / failed call counters is incremented every time.
Number of Established Calls	Indicates the number of established calls. It is incremented as a result of one of the following release reasons if the duration of the call is greater than zero: ▪ GWAPP_REASON_NOT_RELEVANT (0) ▪ GWAPP_NORMAL_CALL_CLEAR (16) ▪ GWAPP_NORMAL_UNSPECIFIED (31) And the internal reasons: ▪ RELEASE_BECAUSE_UNKNOWN_REASON ▪ RELEASE_BECAUSE_REMOTE_CANCEL_CALL ▪ RELEASE_BECAUSE_MANUAL_DISC ▪ RELEASE_BECAUSE_SILENCE_DISC ▪ RELEASE_BECAUSE_DISCONNECT_CODE **Note:** When the duration of the call is zero, the release reason GWAPP_NORMAL_CALL_CLEAR increments the 'Number of Failed Calls due to No Answer' counter. The rest of the release reasons increment the 'Number of Failed Calls due to Other Failures' counter.
Percentage of Successful Calls (ASR)	The percentage of established calls from attempted calls.
Number of Calls Terminated due to a Busy Line	Indicates the number of calls that failed as a result of a busy line. It is incremented as a result of the following release reason: GWAPP_USER_BUSY (17)
Number of Calls Terminated due to No Answer	Indicates the number of calls that weren't answered. It's incremented as a result of one of the following release reasons: ▪ GWAPP_NO_USER_RESPONDING (18) ▪ GWAPP_NO_ANSWER_FROM_USER_ALERTED (19) ▪ GWAPP_NORMAL_CALL_CLEAR (16) (when the call duration is zero)
Number of Calls Terminated due to Forward	Indicates the number of calls that were terminated due to a call forward. The counter is incremented as a result of the following release reason: RELEASE_BECAUSE_FORWARD
Number of Failed Calls due to No Route	Indicates the number of calls whose destinations weren't found. It is incremented as a result of one of the following release reasons: ▪ GWAPP_UNASSIGNED_NUMBER (1) ▪ GWAPP_NO_ROUTE_TO_DESTINATION (3)
Number of Failed Calls due to No Matched Capabilities	Indicates the number of calls that failed due to mismatched device capabilities. It is incremented as a result of an internal identification of capability mismatch. This mismatch is reflected to CDR via the value of the parameter DefaultReleaseReason (default is GWAPP_NO_ROUTE_TO_DESTINATION (3)) or by the

Counter	Description
	GWAPP_SERVICE_NOT_IMPLEMENTED_UNSPECIFIED (79) reason.
Number of Failed Calls due to No Resources	Indicates the number of calls that failed due to unavailable resources or a device lock. The counter is incremented as a result of one of the following release reasons: ▪ GWAPP_RESOURCE_UNAVAILABLE_UNSPECIFIED ▪ RELEASE_BECAUSE_GW_LOCKED
Number of Failed Calls due to Other Failures	This counter is incremented as a result of calls that failed due to reasons not covered by the other counters.
Average Call Duration (ACD) [sec]	The average call duration (ACD) in seconds of established calls. The ACD value is refreshed every 15 minutes and therefore, this value reflects the average duration of all established calls made within a 15 minute period.
Attempted Fax Calls Counter	Indicates the number of attempted fax calls.
Successful Fax Calls Counter	Indicates the number of successful fax calls.

3.5.2.2 Viewing SAS Registered Users

The 'SAS Registered Users' page displays a list of registered users.

➢ **To view the registered users:**

■ Open the 'SAS Registered Users' page (**Status & Diagnostics** tab > **Gateway Statistics** menu > **SAS Registered Users** page item).

Figure 3-135: SAS Registered Users Page

Address Of Record	Contact
<sip:2400@Proxies.ac>	<sip:2400@10.8.210.5>;expires=180
<sip:2401@Proxies.ac>	<sip:2401@10.8.210.5>;expires=180
<sip:2500@Proxies.ac>	<sip:2500@10.8.210.5>;expires=180
<sip:2402@Proxies.ac>	<sip:2402@10.8.210.5>;expires=180
<sip:2403@Proxies.ac>	<sip:2403@10.8.210.5>;expires=180
<sip:2404@Proxies.ac>	<sip:2404@10.8.210.5>;expires=180
<sip:2405@Proxies.ac>	<sip:2405@10.8.210.5>;expires=180

Table 3-39: SAS Registered Users Parameters

Column Name	Description
Address of Record	An address-of-record (AOR) is a SIP or SIPS URI that points to a domain with a location service that can map the URI to another URI (Contact) where the user might be available.
Contact	SIP URI that can be used to contact that specific instance of the User Agent for subsequent requests.

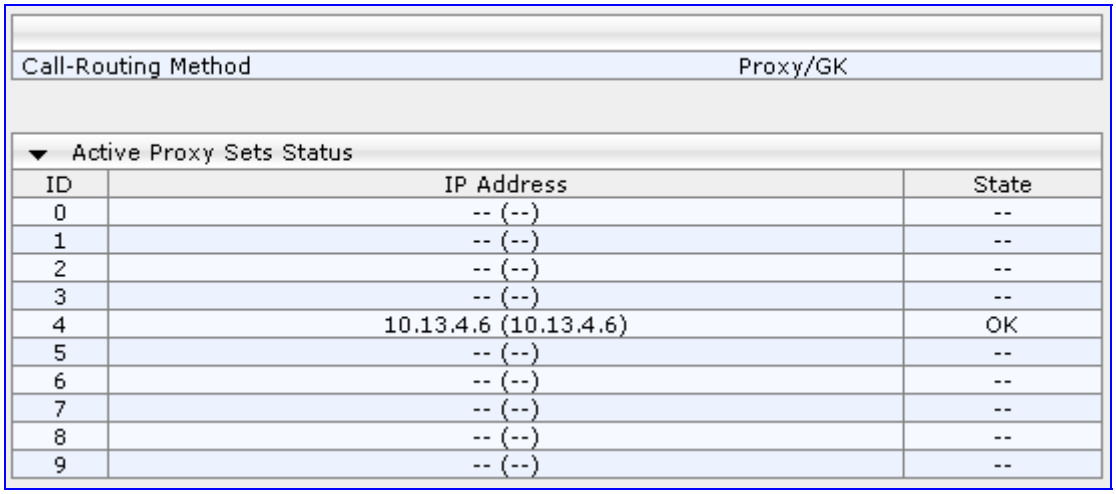
3.5.2.3 Viewing Call Routing Status

The 'Call Routing Status' page provides you with information on the current routing method used by the device. This information includes the IP address and FQDN (if used) of the Proxy server with which the device currently operates.

➢ **To view the call routing status:**

- Open the 'Call Routing Status' page (**Status & Diagnostics** tab > **Gateway Statistics** menu > **Calls Routing Status** page item).

Figure 3-136: Call Routing Status Page



Call-Routing Method	Proxy/GK

Active Proxy Sets Status

ID	IP Address	State
0	-- (--)	--
1	-- (--)	--
2	-- (--)	--
3	-- (--)	--
4	10.13.4.6 (10.13.4.6)	OK
5	-- (--)	--
6	-- (--)	--
7	-- (--)	--
8	-- (--)	--
9	-- (--)	--

Table 3-40: Call Routing Status Parameters

Parameter	Description
Call-Routing Method	▪ Proxy/GK = Proxy server is used to route calls. ▪ Routing Table = The 'Outbound IP Routing Table' is used to route calls.
IP Address	▪ Not Used = Proxy server isn't defined. ▪ IP address and FQDN (if exists) of the Proxy server with which the device currently operates.
State	▪ N/A = Proxy server isn't defined. ▪ OK = Communication with the Proxy server is in order. ▪ Fail = No response from any of the defined Proxies.

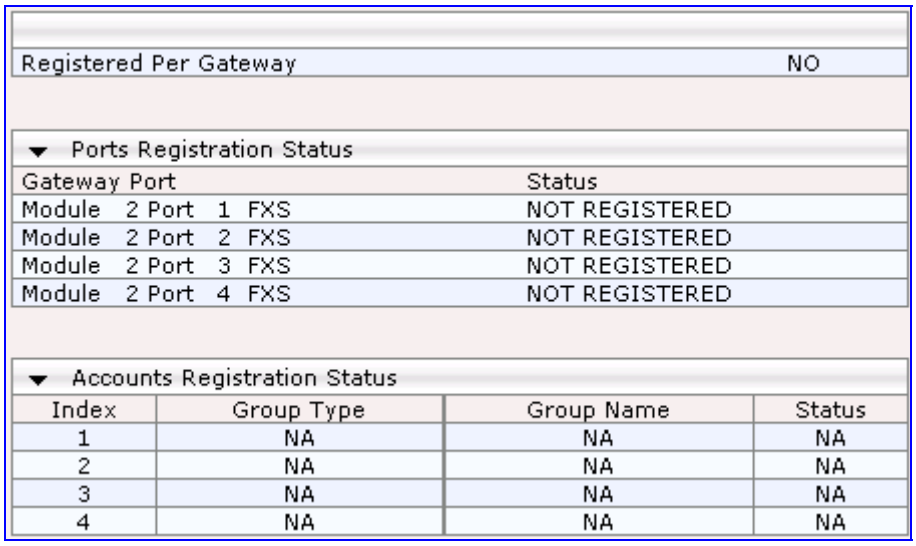
3.5.2.4 Viewing Registration Status

The 'Registration Status' page displays whether the device, endpoints and SIP Accounts are registered to a SIP Registrar/Proxy server.

➢ **To view Registration status:**

- Open the 'Registration Status' page (**Status & Diagnostics** tab > **Gateway Statistics** menu > **Registration Status** page item).

Figure 3-137: Registration Status Page



Registered Per Gateway: NO

Ports Registration Status

Gateway Port	Status
Module 2 Port 1 FXS	NOT REGISTERED
Module 2 Port 2 FXS	NOT REGISTERED
Module 2 Port 3 FXS	NOT REGISTERED
Module 2 Port 4 FXS	NOT REGISTERED

Accounts Registration Status

Index	Group Type	Group Name	Status
1	NA	NA	NA
2	NA	NA	NA
3	NA	NA	NA
4	NA	NA	NA

- **Registered Per Gateway:**
  - 'YES' = registration is per device
  - 'NO' = registration is not per device
- **Ports Registration Status:**
  - 'REGISTERED' = channel is registered
  - 'NOT REGISTERED' = channel not registered
- **Accounts Registration Status:** registration status based on the Accounts table (configured in "Configuring the Account Table" on page 118):
  - **Group Type:** type of served group - Trunk Group or IP Group
  - **Group Name:** name of the served group, if applicable
  - **Status:** indicates whether or not the group is registered ('Registered' or 'Unregistered')

Note: The registration mode (i.e., per device, endpoint, account. or no registration) is configured in the 'Trunk Group Settings' table (refer to "Configuring Trunk Group Settings" on page 104) or using the TrunkGroupSettings *ini* file parameter.

3.5.2.5 Viewing IP Connectivity

The 'IP Connectivity' page displays online, read-only network diagnostic connectivity information on all destination IP addresses configured in the 'Outbound IP Routing Table' page (refer to "Configuring the Outbound IP Routing Table" on page 158).

Notes:

- This information is available only if the parameter 'Enable Alt Routing Tel to IP'/AltRoutingTel2IPMode (refer to "Configuring Routing General Parameters" on page 157) is set to 1 (Enable) or 2 (Status Only).
- The information in columns 'Quality Status' and 'Quality Info' (per IP address) is reset if two minutes elapse without a call to that destination.

➢ **To view the IP connectivity information:**

1. In the 'Routing General Parameters' page, set the parameter 'Enable Alt Routing Tel to IP' (or *ini* file parameter AltRoutingTel2IPEnable) to Enable **[1]** or Status Only **[2]**.
2. Open the 'IP Connectivity' page (**Status & Diagnostics** tab > **Gateway Statistics** menu > **IP Connectivity** page item).

Figure 3-138: IP Connectivity Page

	IP Address	Host Name	Connectivity Method	Connectivity Status	Quality Status	Quality Info	DNS Status
1	Unused	---	---	---	---	---	---
2	Unused	---	---	---	---	---	---
3	Unused	---	---	---	---	---	---
4	Unused	---	---	---	---	---	---
5	Unused	---	---	---	---	---	---
6	Unused	---	---	---	---	---	---
7	Unused	---	---	---	---	---	---
8	Unused	---	---	---	---	---	---
9	Unused	---	---	---	---	---	---
10	Unused	---	---	---	---	---	---

Table 3-41: IP Connectivity Parameters

Column Name	Description
IP Address	The IP address can be one of the following: ▪ IP address defined as the destination IP address in the 'Outbound IP Routing Table'. ▪ IP address resolved from the host name defined as the destination IP address in the 'Outbound IP Routing Table'.
Host Name	Host name (or IP address) as defined in the 'Outbound IP Routing Table'.
Connectivity Method	The method according to which the destination IP address is queried periodically (ICMP ping or SIP OPTIONS request).

Column Name	Description
Connectivity Status	The status of the IP address' connectivity according to the method in the 'Connectivity Method' field. ▪ OK = Remote side responds to periodic connectivity queries. ▪ Lost = Remote side didn't respond for a short period. ▪ Fail = Remote side doesn't respond. ▪ Init = Connectivity queries not started (e.g., IP address not resolved). ▪ Disable = The connectivity option is disabled, i.e., parameter 'Alt Routing Tel to IP Mode' (AltRoutingTel2IPMode *ini*) is set to 'None' or 'QoS'.
Quality Status	Determines the QoS (according to packet loss and delay) of the IP address. ▪ Unknown = Recent quality information isn't available. ▪ OK ▪ Poor **Notes:** ▪ This parameter is applicable only if the parameter 'Alt Routing Tel to IP Mode' is set to 'QoS' or 'Both' (AltRoutingTel2IPMode = 2 or 3). ▪ This parameter is reset if no QoS information is received for 2 minutes.
Quality Info.	Displays QoS information: delay and packet loss, calculated according to previous calls. **Notes:** ▪ This parameter is applicable only if the parameter 'Alt Routing Tel to IP Mode' is set to 'QoS' or 'Both' (AltRoutingTel2IPMode = 2 or 3). ▪ This parameter is reset if no QoS information is received for 2 minutes.
DNS Status	DNS status can be one of the following: ▪ DNS Disable ▪ DNS Resolved ▪ DNS Unresolved

4 INI File Configuration

The device can also be configured by loading an *ini* file containing user-defined parameters. The *ini* file can be loaded to the device using the following methods:

- Web interface (refer to "Backing Up and Restoring Configuration" on page 208)
- AudioCodes' BootP/TFTP utility (refer to the Product Reference Manual)
- Any standard TFTP server

The *ini* file configuration parameters are saved in the device's non-volatile memory when the file is loaded to the device. If a parameter is excluded from the loaded *ini* file, the default value is assigned to that parameter (according to the *cmp* file running on the device), thereby, overriding the value previously defined for that parameter.

Notes:

- For a list and description of the *ini* file parameters, refer to "Configuration Parameters Reference" on page 247.
- Some parameters are configurable only through the *ini* file (and not the Web interface).
- To restore the device to default settings using the *ini* file, refer to "Restoring Factory Default Settings" on page 471.

4.1 INI File Format

The *ini* file can be configured with any number of parameters. These *ini* file parameters can be one of the following parameter types:

- Individual parameters (refer to "Configuring Individual *ini* File Parameters" on page 221)
- Table parameters (refer to "Configuring *ini* File Table Parameters" on page 222)

4.1.1 Configuring Individual *ini* File Parameters

The format of individual *ini* file parameters includes an optional, subsection name (group name) to conveniently group similar parameters by their functionality. Following this line are the actual parameter settings. These format lines are shown below:

```
[subsection name]
; the subsection name is optional.
Parameter Name = Parameter Value
Parameter Name = Parameter Value
; Remark
```

For general *ini* file formatting rules, refer to "General *ini* File Formatting Rules" on page 224.

An example of an *ini* file containing individual *ini* file parameters is shown below:

```
[System Parameters]
SyslogServerIP = 10.13.2.69
EnableSyslog = 1
; these are a few of the system-related parameters.
[Web Parameters]
LogoWidth = '339'
WebLogoText = 'My Device'
UseWeblogo = 1
; these are a few of the Web-related parameters.
[Files]
CallProgressTonesFileName = 'cpusa.dat'
```

4.1.2 Configuring *ini* File Table Parameters

The *ini* file table parameters allow you to configure tables which can include multiple parameters (*columns*) and row entries (*index*). When loading an *ini* file to the device, it's recommended to include only tables that belong to applications that are to be configured (dynamic tables of other applications are empty, but static tables are not).

The *ini* file table parameter is composed of the following elements:

- **Title of the table:** The name of the table in square brackets (e.g., [MY_TABLE_NAME]).
- **Format line:** Specifies the columns of the table (by their string names) that are to be configured.
  - The first word of the Format line must be 'FORMAT', followed by the Index field name and then an equal (=) sign. After the equal sign, the names of the columns are listed.
  - Columns must be separated by a comma (,).
  - The Format line must only include columns that can be modified (i.e., parameters that are not specified as read-only). An exception is Index fields, which are mandatory.
  - The Format line must end with a semicolon (;).

- **Data line(s):** Contain the actual values of the columns (parameters). The values are interpreted according to the Format line.
  - The first word of the Data line must be the table's string name followed by the Index field.
  - Columns must be separated by a comma (,).
  - A Data line must end with a semicolon (;).
- **End-of-Table Mark:** Indicates the end of the table. The same string used for the table's title, preceded by a backslash (\), e.g., [\MY_TABLE_NAME].

The following displays an example of the structure of an *ini* file table parameter.

```
[Table Title]
; This is the title of the table.
FORMAT Index = Column Name1, Column Name2, Column Name3;
; This is the Format line.
Index 0 = value1, value2, value3;
Index 1 = value1, $$, value3;
; These are the Data lines.
[\Table Title]
; This is the end-of-the-table-mark.
```

The *ini* file table parameter formatting rules are listed below:

- Indices (in both the Format and the Data lines) must appear in the same order. The Index field must never be omitted.
- The Format line can include a subset of the configurable fields in a table. In this case, all other fields are assigned with the pre-defined default values for each configured line.
- The order of the fields in the Format line isn't significant (as opposed to the Index fields). The fields in the Data lines are interpreted according to the order specified in the Format line.
- The double dollar sign ($$) in a Data line indicates the default value for the parameter.
- The order of the Data lines is insignificant.
- Data lines must match the Format line, i.e., it must contain exactly the same number of Indices and Data fields and must be in exactly the same order.
- A row in a table is identified by its table name and Index field. Each such row may appear only once in the *ini* file.
- Table dependencies: Certain tables may depend on other tables. For example, one table may include a field that specifies an entry in another table. This method is used to specify additional attributes of an entity, or to specify that a given entity is part of a larger entity. The tables must appear in the order of their dependency (i.e., if Table X is referred to by Table Y, Table X must appear in the *ini* file before Table Y).

For general *ini* file formatting rules, refer to "General *ini* File Formatting Rules" on page 224.

The table below displays an example of an *ini* file table parameter:

```
[ CodersGroup0 ]
FORMAT CodersGroup0 Index = CodersGroup0 Name, CodersGroup0 pTime,
CodersGroup0_rate, CodersGroup0_PayloadType, CodersGroup0_Sce;
CodersGroup0 0 = g711Alaw64k, 20, 0, 255, 0;
CodersGroup0 1 = eg711Ulaw, 10, 0, 71, 0;
CodersGroup0 2 = eg711Ulaw, 10, 0, 71, 0;
[ \CodersGroup0 ]
```

Note: Do not include read-only parameters in the *ini* file table parameter as this can cause an error when attempting to load the file to the device.

4.1.3 General *ini* File Formatting Rules

The *ini* file must adhere to the following format rules:

- The *ini* file name must not include hyphens (-) or spaces; if necessary, use an underscore (_) instead.
- Lines beginning with a semi-colon (;) are ignored. These can be used for adding remarks in the *ini* file.
- A carriage return (i.e., Enter) must be done at the end of each line.
- The number of spaces before and after the equals sign (=) is irrelevant.
- Subsection names for grouping parameters are optional.
- If there is a syntax error in the parameter name, the value is ignored.
- Syntax errors in the parameter's value can cause unexpected errors (parameters may be set to the incorrect values).
- Parameter string values that denote file names (e.g., CallProgressTonesFileName) must be enclosed with inverted commas ('…'), e.g., CallProgressTonesFileName = 'cpt_usa.dat'
- The parameter name is not case-sensitive.
- The parameter value is not case-sensitive, except for coder names.
- The *ini* file must end with at least one carriage return.

4.2 Modifying an *ini* File

You can modify an *ini* file currently used by the device. Modifying an *ini* file instead of loading an entirely new *ini* file preserves the device's current configuration, including factory default values.

➢ **To modify an *ini* file:**

1. Save the current *ini* file from the device to your PC, using the Web interface (refer to "Backing Up and Restoring Configuration" on page 208).
2. Open the *ini* file (using a text file editor such as Microsoft Notepad), and then modify the *ini* file parameters according to your requirements.
3. Save the modified *ini* file, and then close the file.
4. Load the modified *ini* file to the device, using either the BootP/TFTP utility or the Web interface (refer to "Backing Up and Restoring Configuration" on page 208).

Tip: Before loading the *ini* file to the device, verify that the file extension of the *ini* file is correct, i.e., **.ini*.

4.3 Secured Encoded *ini* File

The *ini* file contains sensitive information that is required for the functioning of the device. Typically, it is loaded to or retrieved from the device using TFTP or HTTP. These protocols are not secure and are vulnerable to potential hackers. To overcome this security threat, the AudioCodes' TrunkPack Downloadable Conversion Utility (DConvert) utility allows you to binary-encode the *ini* file before loading it to the device (refer to the *Product Reference Manual)*. If you download an *ini* file from the device to a folder on your PC (using the Web interface - refer to Backing Up and Restoring Configuration) that was initially loaded to the device as encoded, the file is saved encoded and vice versa.

Note: The procedure for loading an encoded *ini* file is identical to the procedure for loading an unencoded *ini* file.

5 Element Management System (EMS)

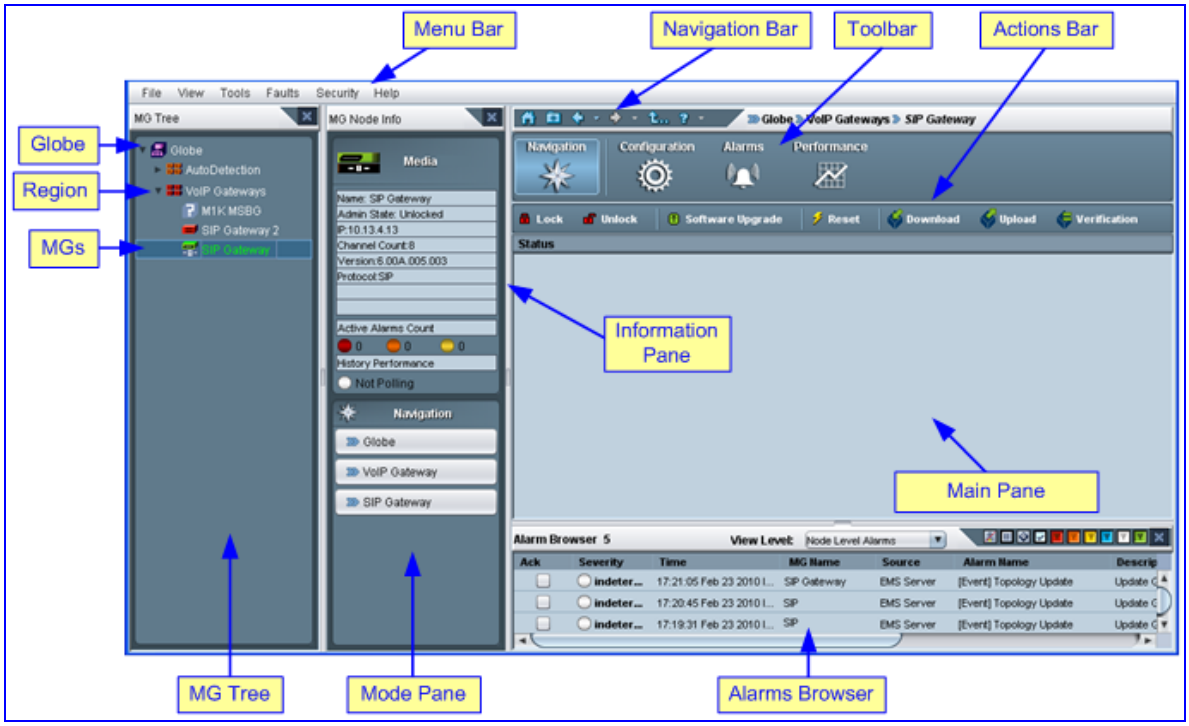
This section provides a brief description on configuring various device configurations using AudioCodes Element Management System (EMS). The EMS is an advanced solution for standards-based management of gateways within VoP networks, covering all areas vital for the efficient operation, administration, management and provisioning (OAM&P) of AudioCodes' families of gateways. The EMS enables Network Equipment Providers (NEPs) and System Integrators (SIs) the ability to offer customers rapid time-to-market and inclusive, cost-effective management of next-generation networks. The standards-compliant EMS uses distributed SNMP-based management software, optimized to support day-to-day Network Operation Center (NOC) activities, offering a feature-rich management framework. It supports fault management, configuration and security.

For a detailed description of the EMS tool, refer to the *EMS User's Manual* and *EMS Server IOM Manual*.

5.1 Familiarizing yourself with EMS GUI

The areas of the EMS graphical user interface (GUI) are shown in the figure below:

Figure 5-1: Areas of the EMS GUI



The MG Tree is a hierarchical tree-like structure that lists all the devices managed by EMS. The tree includes the following icons:

- **Globe** : highest level in the tree from which a Region can be added.
- **Region** : defines a group (e.g., geographical location) to which devices can be added. If you click a Region that is defined with devices (MG's), the Main pane (see figure above) displays a list of all the devices pertaining to the Region.
- **MG** : defines the device. This is the lowest level in the tree. If you click an **MG** icon, the Main pane (see figure above) displays a graphical representation of the device's chassis.

5.2 Securing EMS-Device Communication

5.2.1 Configuring IPSec

Before you can configure the device through the EMS, you need to configure the secure communication protocol IPSec for communicating between the EMS and the device. Before you enable IPSec in the EMS, you must define the IPSec IKE pre-shared key in a secure manner. This is performed through an SSH secure shell client session (e.g. PuTTY). Once you have defined the IPSec IKE pre-shared key, you must enter the same IPSec IKE pre-shared key in the EMS when you define the device.

Before performing the procedure below, ensure that you have the following information:

- The IP address of the EMS Server that is to communicate with the device
- An initial password for the IKE pre-shared key

Notes:

- The device is shipped with SSH enabled.
- The configuration text is case- and space-sensitive. Type the text rather than copy-and-paste. Save the IKE pre-shared key as later on you need to enter the same value in the EMS when defining the device.
- For more information on CLI, refer to the *Product Reference Manual.*
- For more information on securing communication protocols, refer to the *EMS Users Manual*.

➢ **To configure the device for communicating via IPSec with the EMS:**

1. Open an SSH Client session (e.g. PuTTY), and then connect to the device.
   - If a message appears with the RSA host key, click **Yes** to continue.
   - The default username and password are "Admin" (case-sensitive). Verify that the shell prompt appears ("\> ").
2. Type **Conf**, and then press Enter.

   ```
   /CONFiguration>
   ```
3. Type **cf set**, and then press Enter; the following prompt is displayed:

   ```
   Enter data below. Type a period (.) on an empty line to
   finish.
   ```

   The configuration session is now active and all data entered at the terminal is parsed as configuration text (formatted as an *ini* file).
4. Type the following at the configuration session:

```
[ IPsecSATable ]
FORMAT IPsecSATable Index =
IPsecSATable RemoteEndpointAddressOrName,
IPsecSATable_AuthenticationMethod, IPsecSATable_SharedKey,
IPsecSATable SourcePort, IPsecSATable DestPort,
IPsecSATable Protocol, IPsecSATable Phase1SaLifetimeInSec,
IPsecSATable_Phase2SaLifetimeInSec,
IPsecSATable Phase2SaLifetimeInKB, IPsecSATable DPDmode,
IPsecSATable IPsecMode, IPsecSATable RemoteTunnelAddress,
IPsecSATable RemoteSubnetIPAddress,
```

```
IPsecSATable RemoteSubnetPrefixLength;
IPsecSATable 1 = <IP address>, 0, <IKE password>, 0, 0, 0, 28800,
28800, 0, 0, 0, 0.0.0.0, 0.0.0.0, 16
[ \IPsecSATable ]
EnableIPSec = 1
```

where:

- <IKE password> is the password for the initial IKE pre-shared key.
- <IP address> is the IP address of the EMS server used for connecting to the device for which IPSec connectivity is established.

5. To end the PuTTY configuration session, type a full-stop (".") on an empty line; the device responds with the following:

```
INI File replaced
```

6. To save the configuration to the non-volatile memory, type **sar**; the device reboots with IPSec enabled.

Note: If you have enabled IPSec and you want to change the IP address and/or IKE password, you need to first disable IPSec. Perform the procedure as above, but omit the lines **[IPsecSATable**], and set EnableIPSec to 0. Once you have done this, repeat the exact procedure as described above, but with the new IP address and/or password.

5.2.2 Changing SSH Login Password

For security, it is recommended to change the default SSH Client login password, using the SSH client.

➢ **To change the SSH login password:**

1. Open an SSH Client session (e.g. PuTTY), and then connect, using the default user name and password ("Admin" - case sensitive), to the device. If a message appears with the RSA host key, click **Yes** to continue; the shell prompt appears ("\> ").
2. At the CLI prompt, type the command **chpw** and specify the existing and new passwords.

```
chpw <old_password> <new_password>
```

where:

- <old_password> is the existing password
- <new_password> is the new password

The device responds with the message "Password changed".

3. Close the SSH client session and reconnect using the new password.

Note: The default user name ("Admin") cannot be changed from within an SSH client session.

5.3 Adding the Device in EMS

Once you have defined the IPSec communication protocol for communicating between EMS and the device and configured the device's IP address (refer to the device's *Installation Manual*), you can add the device in the EMS.

Adding the device to the EMS includes the following main stages:

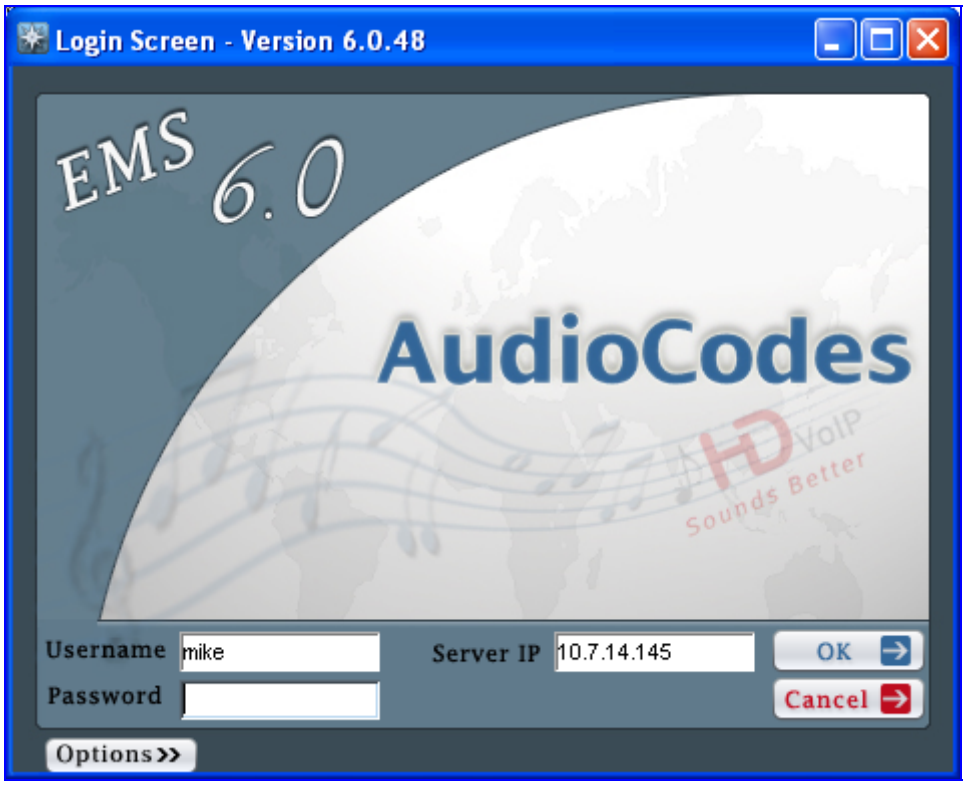
a. Adding a Region

b. Defining the device's IP address (and other initial settings)

➢ **To initially setup the device in EMS:**

1. Start the EMS by double-clicking the shortcut icon on your desktop, or from the **Start** menu, point to **Programs**, point to **EMS Client**, and then click **EMS CLient**; the Login Screen appears:

Figure 5-2: EMS Login Screen



2. Enter your login username and password, the EMS server's IP address, and then click **OK**.

3. Add a Region for your deployed device, by performing the following:

a. In the MG Tree, right-click the **Globe** icon, and then click **Add Region**; the Region dialog box appears.

Figure 5-3: Adding a Region

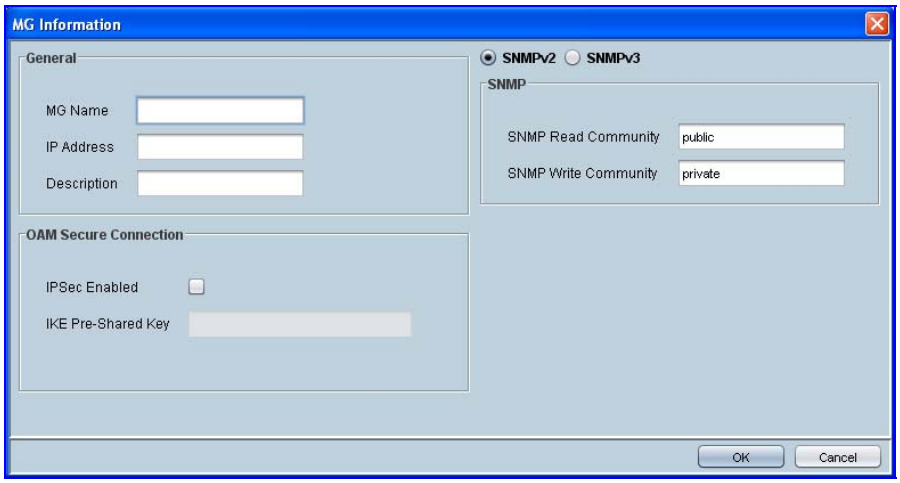
b. In the 'Region Name' field, enter a name for the Region (e.g., a geographical name), and then click **OK**; the Region is added to the MG Tree list.

4. Verify that the device is up and running (by performing a ping to its IP address).

5. Add the device to the Region, by performing the following:

a. Right-click the added Region icon, and then from the shortcut menu, choose **Add MG**; the MG Information dialog box appears.

Figure 5-4: Defining the IP Address



b. Enter an arbitrary name for the device, and then in the 'IP Address' field, enter the device's IP address

c. Ensure that 'IPSec Enabled' check box is selected, and then enter the IPSec Preshared Key (defined in Configuring IPSec on page 228).

d. Click **OK**; the device is added to the Region and appears listed in the MGs List.

Note: The Pre-shared Key string defined in the EMS must be identical to the one that you defined for the device. When IPSec is enabled, default IPSec/IKE parameters are loaded to the device.

5.4 Configuring Trunks

This section describes the provisioning of trunks:

- E1/T1Trunk configuration (refer to "General Trunk Configuration" on page 232)
- ISDN NFAS (refer to Configuring ISDN NFAS on page 233)

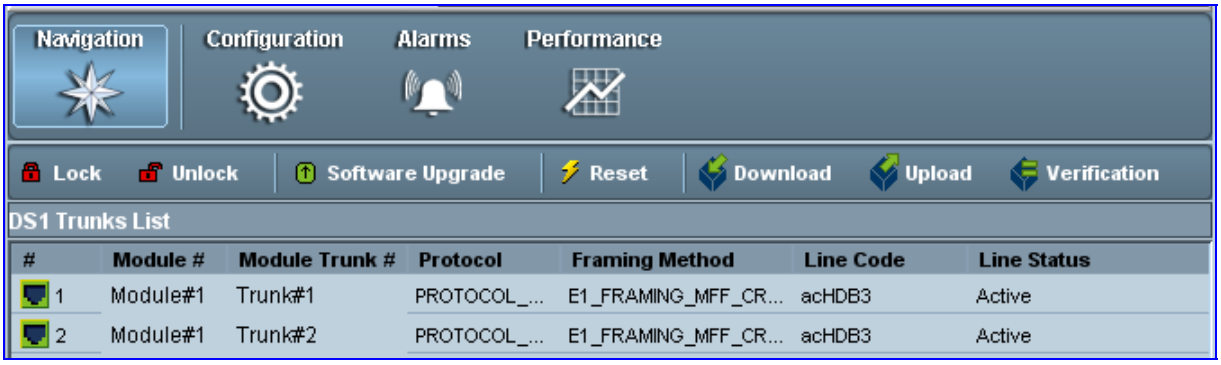
5.4.1 General Trunk Configuration

This section describes how to provision a PSTN trunk.

➢ **To provision a trunk:**

1. In the MG Tree, select the required device; the device's graphical display is shown in the Main pane.
2. Click the Trunk module; the DS1 Trunks List appears.

Figure 5-5: DS Trunks List



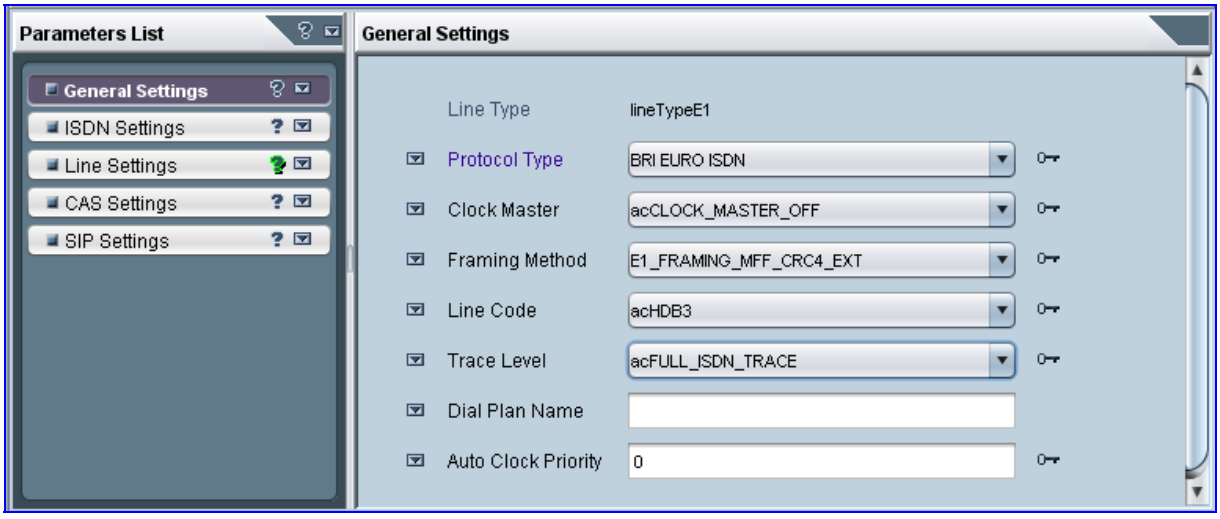
3. Click a trunk, and then from the MG Mode pane, select the **PSTN** menu, and then the **Trunks Channels** menu; the Trunks Channels Table appears in the Main pane.

Figure 5-6: Trunks Channels Table

Navigation
Configuration
Alarms
Performance
Trunks Channels Table
Module # Module Trunk # PSTN Status 0 1 2 3 4 5 6 7 8 9
1 Module#1 Trunk#1 D_Channel alarm
2 Module#1 Trunk#2 D_Channel alarm

4. Click a trunk and then click the **Configuration** icon; the Trunk SIP Provisioning screen is displayed with the **General Settings** tab selected.

Figure 5-7: General Settings Screen



5. From the 'Protocol Type' drop-down list, select the required protocol.
6. From the 'Framing Method' drop-down list, select the required framing method. For E1, always set this parameter to Extended Super Frame.
7. From the 'Clock Master' drop-down list, set the Clock Master to one of the following values:
   - Clock Master OFF: the Clock Source is recovered from the Trunk line.
   - Clock Master ON: the Clock Source is provided by the internal TDM bus clock source, according to the parameter TDM Bus Clock Source.
8. Select the other tabs to continue configuring the PSTN trunks.

Notes:

- When changing 'Protocol Type' from 'None' to any other protocol, reset the device. You're not required to reset the device when making subsequent changes to 'Protocol Type'.
- Most Trunk provisioning parameters require that a Trunk Lock / Unlock be performed before / after configuring them. When performing a Lock action, all active calls are dropped and users cannot make new calls. This is Trunk Out Of Service mode.
- Upon initial configuration, do not change the Admin State of the trunks to unlock (it is changed automatically after the device is reset in EMS).

5.4.2 Configuring ISDN NFAS

This section describes how to configure ISDN-NFAS trunks as an initial configuration.

In regular T1 ISDN trunks, a single 64 kbps channel carries signaling for the other 23 B-channels of that particular T1 trunk. This channel is called the D-channel and usually resides on timeslot #24. The ISDN Non-Facility Associated Signaling (NFAS) feature enables the use of a single D-channel to control multiple PRI interfaces.

With NFAS it is possible to define a group of T1 trunks, called an NFAS group, in which a single D-channel carries ISDN signaling messages for the entire group. The NFAS group's B-channels are used to carry traffic such as voice or data. The NFAS mechanism also enables definition of a backup D-channel on a different T1 trunk, to be used if the primary D-channel fails.

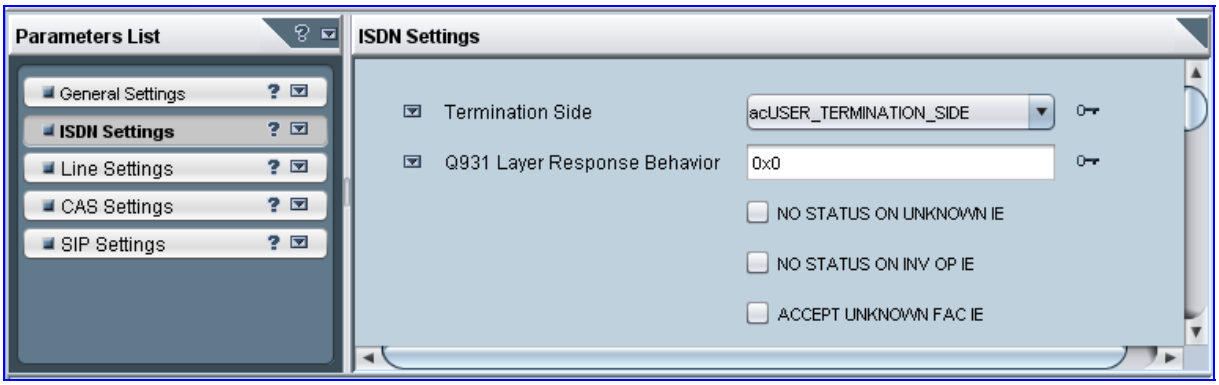
The NFAS group can comprise up to 10 T1 trunks. Each T1 trunk is called an 'NFAS member'. The T1 trunk whose D-channel is used for signaling is called the 'Primary NFAS Trunk'. The T1 trunk whose D-channel is used for backup signaling is called the 'Backup NFAS Trunk'. The primary and backup trunks each carry 23 B-channels while all other NFAS trunks each carry 24 B channels.

ISDN-NFAS Trunks can be configured offline or on-the-fly.

➢ **To configure ISDN-NFAS Trunks offline:**

1. In the MG Tree, select the required device; the device's graphical display is shown in the Main pane.
2. Click the Trunk module; the DS1 Trunks List appears.
3. Click a trunk, and then from the MG Mode pane, select the **PSTN** menu, and then the **Trunks Channels** menu; the Trunks Channels Table appears in the Main pane.
4. Click a trunk channel and then click the **Configuration** icon; the Trunk SIP Provisioning screen is displayed with the **General Settings** tab selected.
5. Select the **ISDN Settings** tab; the 'ISDN Settings' screen appears.

Figure 5-8: EMS ISDN Settings Screen



6. Perform the following configurations:
   a. Configure each trunk in the group with the same values for the 'Termination Side' parameter.
   b. Select the 'EXPLICIT INTERFACE ID' check box to configure the Interface ID (see Step d) of a NFAS Trunk. If this field is not set, only the Trunk ID is recognized.
   c. From the 'D-Channel Configuration' drop-down list, select 'Primary NFAS Trunk' for the T1 trunk whose D-channel is used for signaling or 'Backup NFAS Trunk' for the T1 trunk whose D-channel is used for backup signaling. The primary and backup trunks each carry 23 B-channels while all other NFAS trunks each carry 24 B-channels.
   d. In the 'ISDN NFAS Interface ID' field, enter the Interface ID (0 - 255) of the trunk in the NFAS group.
   e. In the 'Group Number' field, enter the device's NFAS Group Number. If this field is set to 0, the trunk is not an NFAS trunk.
   f. Click **Apply**.

g. To apply the configured fields to multiple trunks, use the Profiles that appear on the lower part of the screen.

7. Select the **General Settings** tab, and then configure each trunk in the group with the same values for the following parameters:
   - Protocol Type
   - Framing Method
   - Line Code
8. Burn and reset the device after all the trunks have been configured.

Note: All trunks in the group must be configured with the same values for trunk parameters TerminationSide, ProtocolType, FramingMethod and LineCode.

The procedure below describes how to configure ISDN-NFAS trunks on-the-fly. The configuration process is the same as the initial Offline configuration, but the sequence of configuring or locking the trunks is important.

➢ **To configure ISDN-NFAS Trunks on-the-fly:**

- Unlocking an NFAS Group:
  a. If there is a Backup trunk for this group, it must be unlocked first.
  b. The Primary trunk must be unlocked before unlocking any NFAS trunks.
  c. NFAS trunks should then be unlocked.
- Locking and Removing an NFAS Group:
  a. Lock all NFAS trunks, change their Protocol Type to NONE and then unlock them.
  b. Lock the Backup trunk if it exists. Change its Protocol Type to NONE and then unlock it.
  c. Lock the Primary trunk, change its Protocol Type to NONE and then unlock it.

Note: You cannot re-configure an NFAS group after locking it. You must first set all trunks to Protocol Type NONE and then start configuration again.

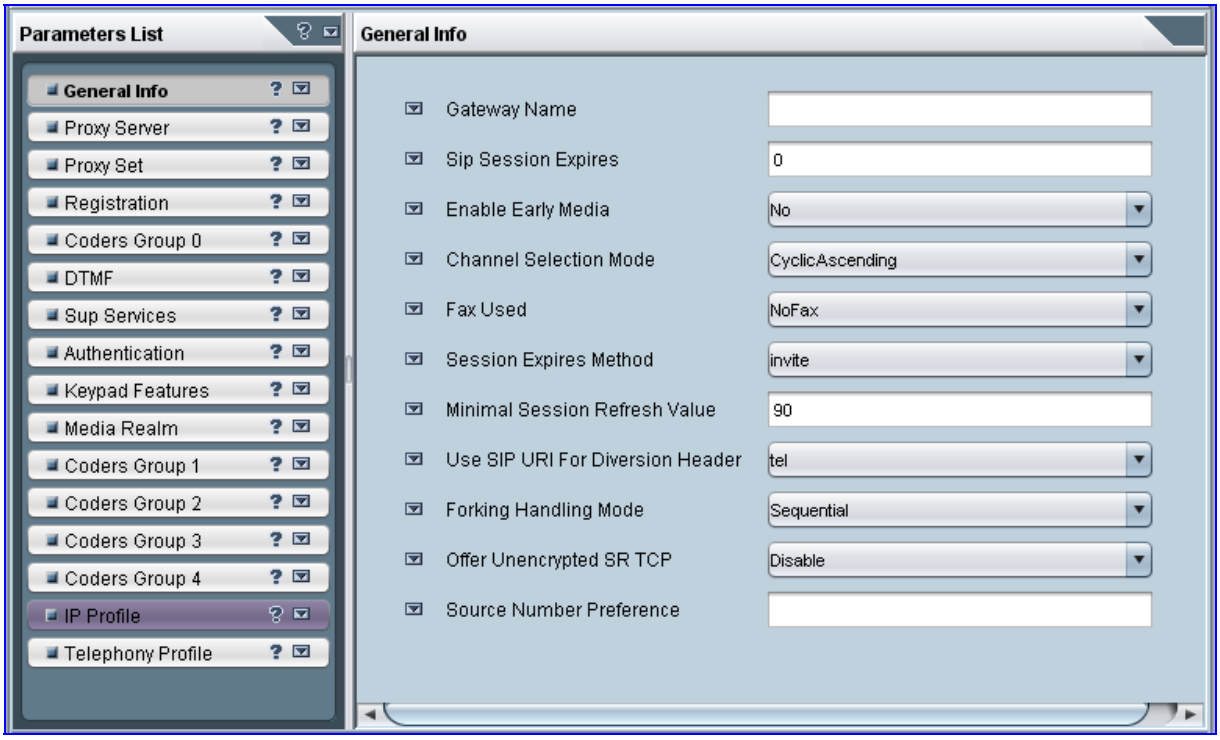
5.5 Configuring Basic SIP Parameters

This section describes how to configure the device with basic SIP control protocol parameters using the EMS.

➢ **To configure basic SIP parameters:**

1. In the MG Tree, select the device that you want to configure; a graphical representation of the device is displayed in the main pane.
2. Open the 'SIP Protocol Definitions' frame (**Configuration** icon > **SIP Protocol Definitions** menu).

Figure 5-9: SIP Protocol Definitions Frame



3. Select the **Coders Group 0** tab; the Coders screen is displayed.
   a. Click the button to add a new Coder entry, and then click **Yes** to confirm.
   b. Double-click each field to enter values.
   c. Right-click the new entry, and then choose **Unlock Rows**.
4. Select the **Proxy Server** tab.
   a. Set 'Proxy Used' to Yes.
   b. (Optional) In the 'Proxy Name' field, enter the Proxy's name. The Proxy name replaces the Proxy IP address in all SIP messages. This means that messages are still sent to the physical Proxy IP address, but the SIP URI contains the Proxy name instead. When no Proxy is used, the internal routing table is used to route the calls.
   c. Click the button, and then click **Yes** to confirm.
   d. Enter the IP address of the Proxy Server.
   e. Right-click the new entry, and then choose **Unlock Rows**.

5. Select the **Registration** tab.
   a. Configure 'Is Register Needed' field:
      - No = the device doesn't register to a Proxy/Registrar server (default).
      - Yes = the device registers to a Proxy/Registrar server at power up and every user-defined interval ('Registration Time' parameter).
   b. Click **Apply** and close the active window.
6. Open the 'SIP EndPoints' frame (**Configuration** icon > **SIP Endpoints** menu).
   a. Click the button to add a new entry, and then click **Yes** to confirm; the 'Phones' screen is displayed.
   b. Double-click each field to enter values.
   c. Right-click the new entry, and then select **Unlock Rows.**
   d. Click **Apply** and close the active window.

Note: For T1 ISDN spans, configure 1-23 (and not 1-24) for B-channels. Channel 24 is a signaling ISDN channel.

7. If a Proxy Server is not implemented, map outgoing telephone calls to IP addresses. Open the 'SIP Routing' frame (**Configuration** icon > **SIP Routing** menu).
8. Select the **Tel to IP** tab.
   a. Click the button to add a new entry, and then click **Yes** to confirm; the Tel to IP Routing table is displayed.
   b. Double-click each field to enter values.
   c. Right-click the new entry and select **Unlock Rows**.
   d. Click **Apply** and close the active window.

5.6 Configuring Advanced IPSec/IKE Parameters

After you have pre-configured IPSec via SSH (refer to "Securing EMS-Device Communication" on page 228), you can optionally configure additional IPSec and IKE entries for other SNMP Managers aside from the EMS.

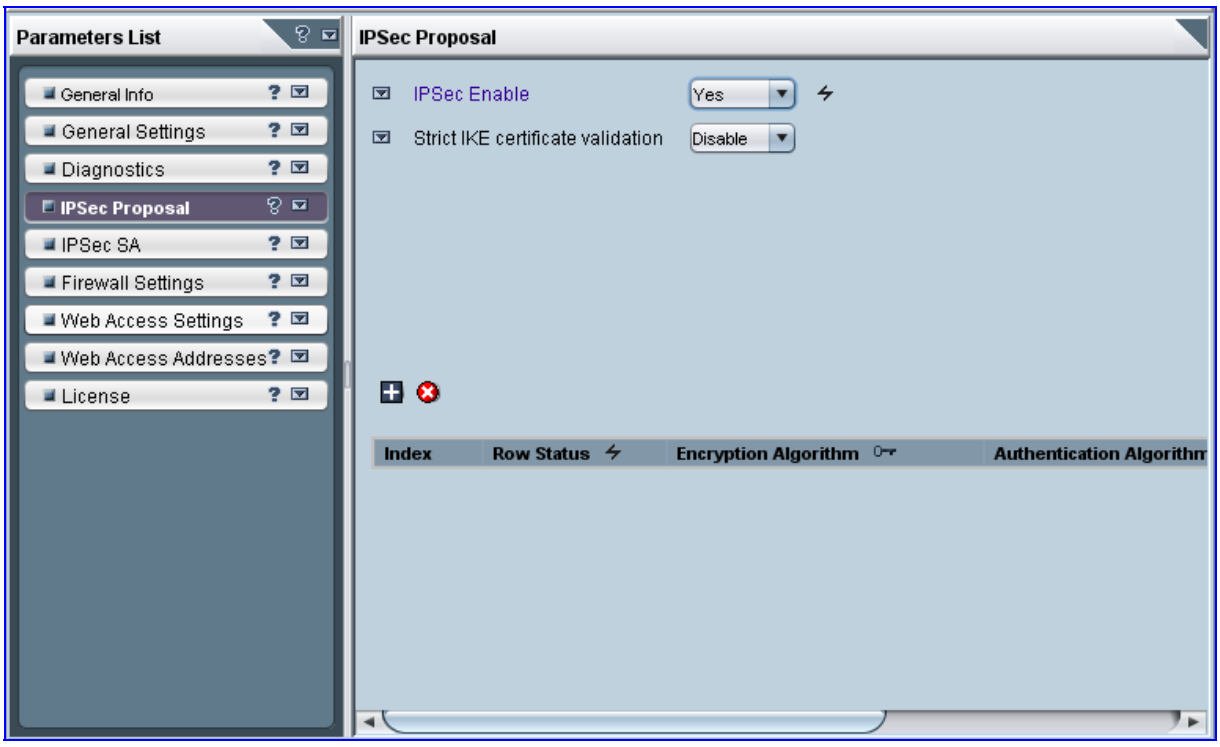
Note: Do not remove the default IPSec and IKE tables that were previously loaded to the device when you enabled IPSec.

➢ **To configure IPSec/IKE tables:**

1. In the MG Tree, select the device.
2. Open the 'MG Info and Security Provisioning' screen (**Configuration** icon > **Info & Security Frame** menu).

3. Select the **IPSec Proposal** tab; the 'IPSec Proposal' screen is displayed.

Figure 5-10: IPSec Table Screen



4. Select the button to add a new entry, and then click **Yes** at the confirmation prompt; a row is added to the table.
5. Enter the required values.
6. Right-click the new entry, and then from the shortcut menu, choose **Unlock rows**.
7. Click **Save**, and then **Close**.
8. Select the **IPSec SA** tab; the 'IPSec SA' screen appears.
9. Repeat steps 4 through 7.

5.7 Provisioning SIP SRTP Crypto Offered Suites

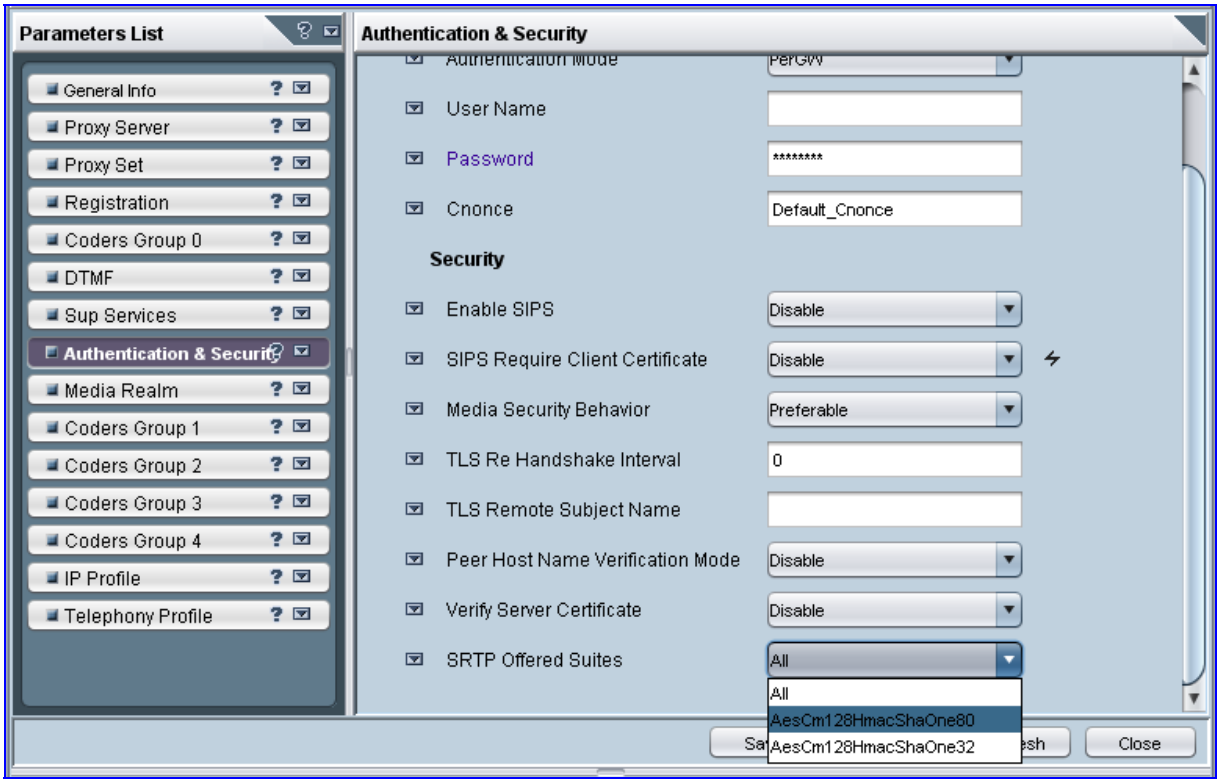
This section describes how to configure offered SRTP crypto suites in the SDP.

➢ **To configure SRTP crypto offered suites:**

1. In the MG Tree, select the device that you want to configure; a graphical representation of the device is displayed in the main pane.

2. Open the 'Authentication & Security' screen (**Configuration** icon > **SIP Protocol Definitions** menu > **Authentication & Security** tab).

Figure 5-11: Authentication & Security Screen



3. From the 'SRTP Offered Suites' (SRTPofferedSuites) drop-down list, select one of the crypto suites.

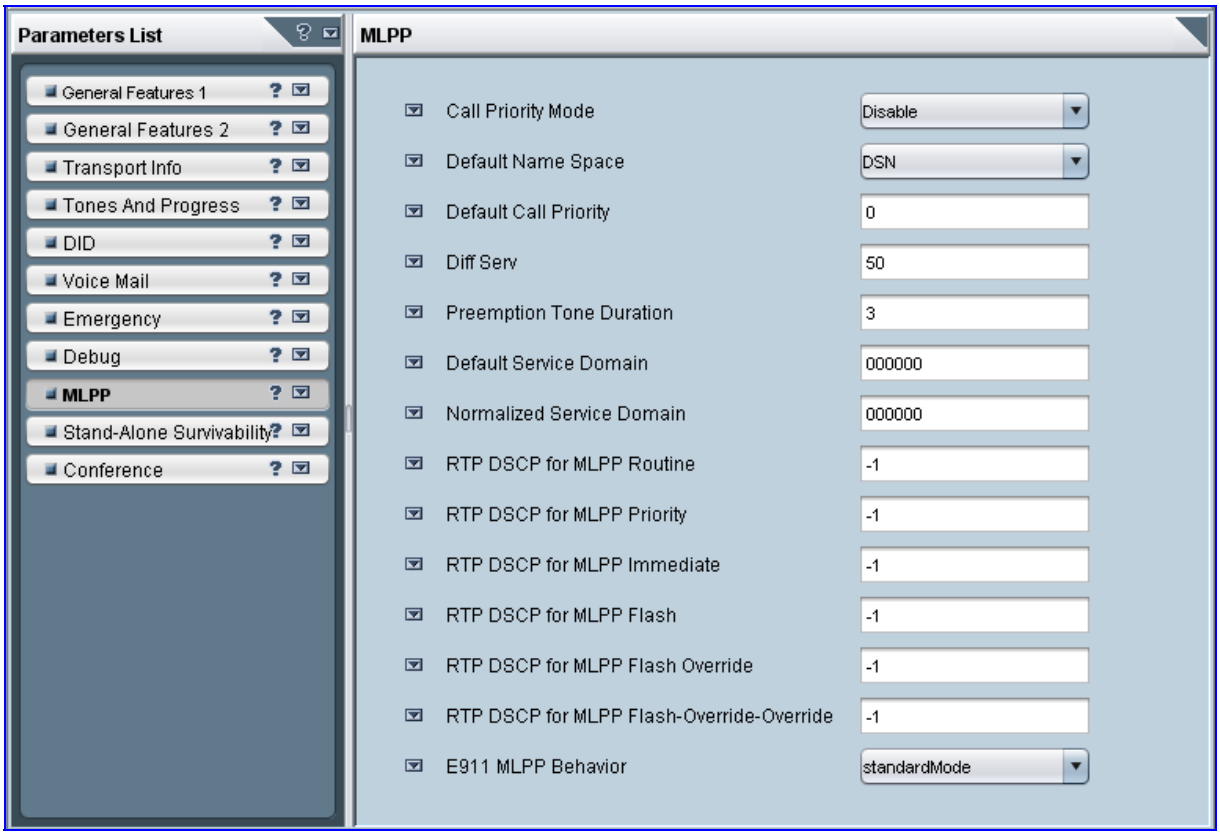
5.8 Provisioning SIP MLPP Parameters

This section describes how to configure the MLPP (Multi-Level Precedence and Preemption) parameters using the EMS.

➢ **To configure the MLPP parameters:**

1. In the MG Tree, select the device that you want to configure; a graphical representation of the device is displayed in the main pane.
2. Open the 'MLPP' screen (**Configuration** icon > **SIP Advanced Configuration** menu> **MLPP** tab).

Figure 5-12: MLPP Screen



3. Configure the MLPP parameters as required.

Note: If the following RTP DSCP parameters are set to "-1" (i.e., Not Configured, Default), the DiffServ value is set with the PremiumServiceClassMediaDiffserv global gateway parameter, or by using IP Profiles: MLPPRoutineRTPDSCP, MLPPPriorityRTPDSCP, MLPPImmediateRTPDSCP, MLPPFlashRTPDSCP, MLPPFlashOverRTPDSCP, MLPPFlashOverOverRTPDSCP, MLPPNormalizedServiceDomain.

5.9 Configuring the Device to Operate with SNMPv3

This section describes the SNMPv3 configuration process:

- Configuring SNMPv3 using SSH
- Configuring SNMPv3 using EMS (non-configured SNMPv3 System)
- Configuring SNMPv3 using EMS (pre-configured SNMPv3 System)

Note: After configuring SNMPv3, ensure that you disable IPSec.

5.9.1 Configuring SNMPv3 using SSH

The procedure below describes how to configure SNMPv3 using SSH.

➢ **To configure the device to operate with SNMPv3 via SSH:**

1. Open an SSH Client session (e.g. PuTTY), and then connect, using the default user name and password ("Admin" - case sensitive) to the device. If a message appears with the RSA host key, click "Yes" to continue. Verify that the shell prompt appears ("\> ").
2. Type **Conf**, and then press Enter.

   ```
   /CONFiguration>
   ```

3. Type **cf set**, and then press Enter; the following prompt is displayed:

   ```
   Enter data below. Type a period (.) on an empty line to
   finish.
   ```

   The configuration session is now active and all data entered at the terminal is parsed as configuration text (formatted as an *ini* file).
4. Type the following text at the configuration session:

   ```
   [ SNMPUsers ]
   FORMAT SNMPUsers_Index = SNMPUsers_Username,
   SNMPUsers_AuthProtocol, SNMPUsers_PrivProtocol,
   SNMPUsers_AuthKey, SNMPUsers_PrivKey, SNMPUsers_Group;
   SNMPUsers 0 = v3user, 2, 1,<auth password>,<priv password>, 1;
   [ \SNMPUsers ]
   ```

   where:

   - <auth password> is the password for the for the authentication protocol
   - <priv password> is the password for the privacy protocol

   Possible values for AuthProtocol:

   - 0 – none
   - 1 - MD5
   - 2 - SHA-1

   Possible values for PrivProtocol:

   - 0 – none
   - 1 – DES
   - 3 - AES128
5. To end the PuTTY configuration session, type a full-stop (".") on an empty line; the device responds with the following:

   ```
   INI File replaced
   ```

6. To save the configuration to the non-volatile memory, type **sar**; the device reboots with IPSec enabled.

5.9.2 Configuring EMS to Operate with a Pre-configured SNMPv3 System

The procedure below describes how to configure the device with a pre-configured SNMPv3.

➢ **To configure the EMS to operate with a pre-configured SNMPv3 system:**

1. In the MG Tree, select the required Region to which the device belongs, and then right-click the device.
2. From the shortcut menu, choose **Details**; the 'MG Information' screen appears.

Figure 5-13: MG Information Screen

3. Select the **SNMPv3** option, configure the SNMP fields, and then click **OK**.
4. Open the 'SNMPv3 Users' screen (**Configuration** icon > **Network Frame** menu > **SNMPv3 Users** tab).
5. From the **SNMPv3 Users** tab's drop-down list, choose **Unit value**; the 'SNMPv3 Users' table is refreshed with the values that you entered in Step 3.
6. Click the **Save** button; the EMS and the device are now synchronized.

5.9.3 Configuring SNMPv3 to Operate with Non-Configured SNMPv3 System

The procedure below describes how to configure SNMPv3 using the EMS.

➢ **To configure the device to operate with SNMPv3 via EMS (to a non-configured System):**

1. In the MG Tree, select the required Region to which the device belongs; the device is displayed in the Main pane.
2. Right-click the device, and then from the shortcut menu, point to **Configuration**, and then click **SNMP Configuration**; the 'SNMP Configuration' window appears.

Figure 5-14: SNMP Configuration Screen

3. Select the **SNMPv3** option.
4. Configure the SNMPv3 fields, and then select the **Update Media Gateway SNMP Settings** check box.
5. Click **OK**; the update progress is displayed.
6. Click **Done** when complete.
7. Open the 'SNMPv3 Users' screen (**Configuration** icon > **Network Frame** menu > **SNMPv3 Users** tab).
8. From the **SNMPv3 Users** tab's drop-down list, choose **Unit value**; the 'SNMPv3 Users' table is refreshed with the values that you entered in Step 4.
9. Click the **Save** button; the EMS and the device are now synchronized.

5.9.4 Cloning SNMPv3 Users

According to the SNMPv3 standard, SNMPv3 users on the SNMP Agent (on the device) cannot be added via the SNMP protocol, e.g. SNMP Manager (i.e., the EMS). Instead, new users must be defined by User Cloning. The SNMP Manager creates a new user according to the original user permission levels.

➢ **To clone SNMPv3 Users:**

1. Open the 'SNMPv3 Users' screen (**Configuration** icon > **Network Frame** menu > **SNMPv3 Users** tab).
2. Select the user with which you wish to clone permission levels.
3. Click the button; the 'New SNMPv3 User' window appears.
4. Provide a new user name, old passwords of the user you clone permissions from and new user passwords.
5. Select a User permission group.
6. If the new user wishes to receive traps to the user-defined destination, select the **Use SNMPv3 User Security Profile for Trap Forwarding** option to provision Trap destination IP and Port. EMS adds this new user to the SNMP Trap Managers Table. It is also possible to define an additional trap destination after a new user is defined.

5.10 Resetting the Device

When you have completed configuring the device, you need to save your settings to the device's flash memory and reset the device.

➢ **To save configuration and reset the device:**

1. In the MG Tree, select the device that you want to reset.
2. On the Actions bar, click the **Reset** button.

Figure 5-15: Confirmation for Saving Configuration and Resetting Device

3. Ensure that the option **Burn Configuration into flash memory** is selected.
4. Click **Yes**; the progress of the reset process is displayed.
5. Click **Done** when complete.

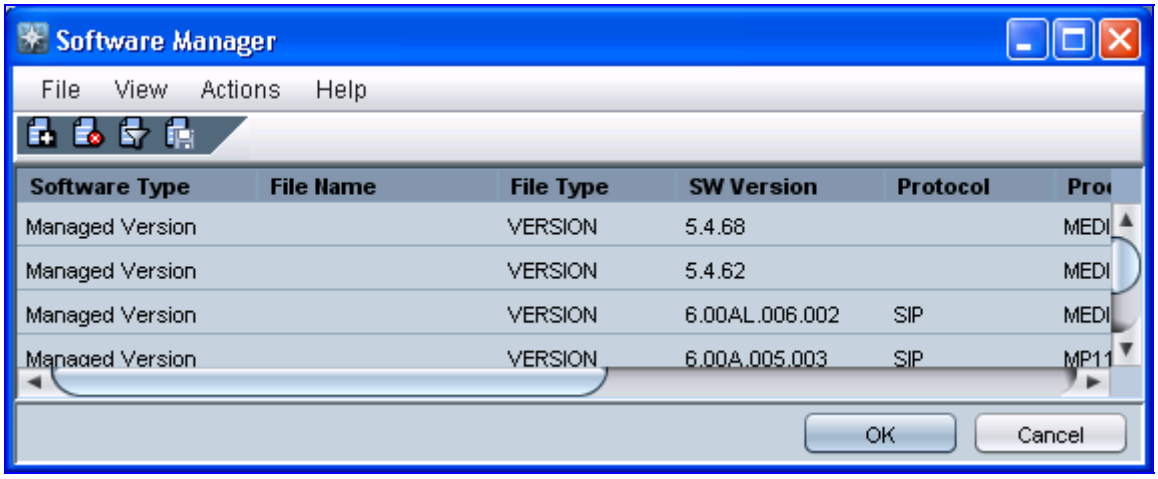
5.11 Upgrading the Device's Software

The procedure below describes how to upgrade the devices software (i.e., cmp file) using the EMS.

➢ **To upgrade the device's cmp file:**

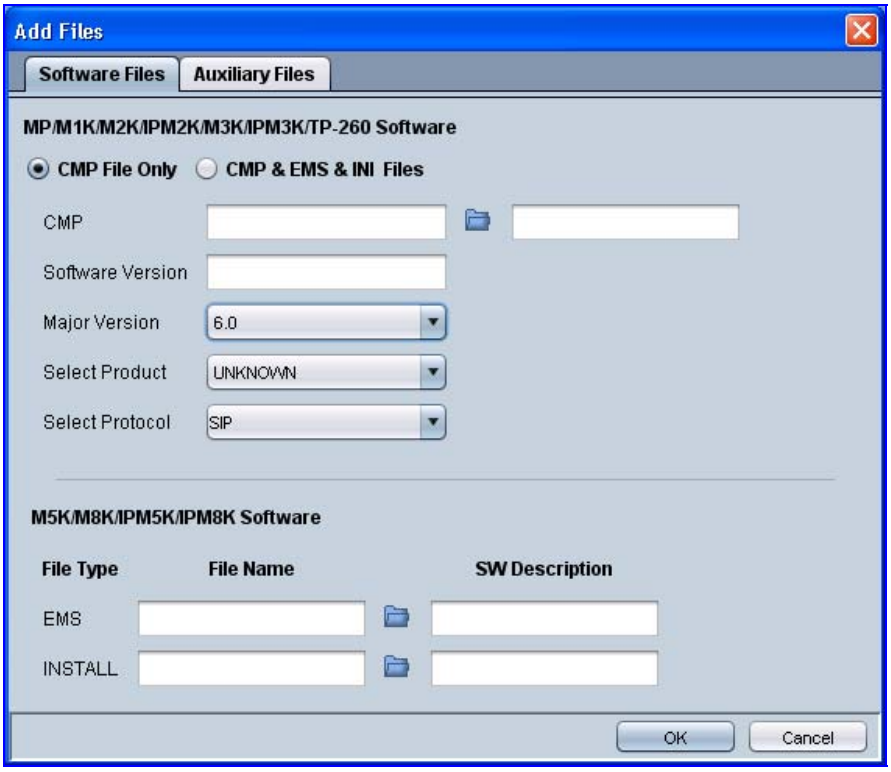
1. From the **Tools** menu, choose **Software Manager**; the 'Software Manager' screen appears.

Figure 5-16: Software Manager Screen


Software Manager
File View Actions Help
Software Type File Name File Type SW Version Protocol Pro
Managed Version VERSION 5.4.68 MEDI
Managed Version VERSION 5.4.62 MEDI
Managed Version VERSION 6.00AL.006.002 SIP MEDI
Managed Version VERSION 6.00A.005.003 SIP MP11
OK
Cancel

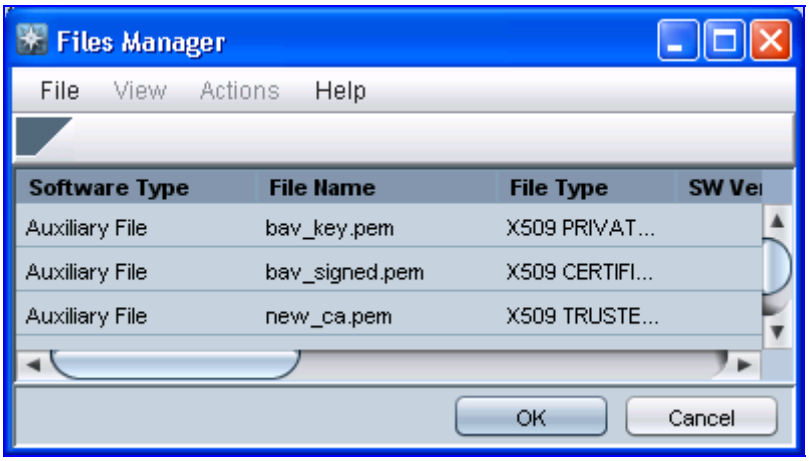
2. Click the **Add File** icon; the 'Add Files' dialog box appears.

Figure 5-17: Add Files Screen



3. Select the cmp file, by performing the following:
   a. Ensure that the **CMP File Only** option is selected.
   b. In the 'CMP' field, click the browse button and navigate to the required cmp file; the software version number of the selected file appears in the 'Software Version' field.
   c. From the 'Major Version' drop-down list, select the version number of the cmp file.
   d. From the 'Select Product' drop-down list, select the type of device.
   e. From the 'Select Protocol' drop-down list, select the the control protocol (i.e., SIP).
4. Click **OK**.
5. In the MG Tree, select the device that you want to upgrade.
6. On the Actions bar, click the **Software Upgrade** button; the 'Files Manager' screen appears.

Figure 5-18: Files Manager Screen



7. Select the file that you want to download to the device, and then click **OK**; a confirmation box appears.
8. Click **Yes** to confirm download; the 'Software Download' screen appears, displaying the download progress.
9. Click **Done** when download is completed successfully.

6 Configuration Parameters Reference

The device's configuration parameters, default values, and their descriptions are documented in this section.

Parameters and values enclosed in square brackets ([...]) represent the *ini* file parameters and their enumeration values; parameters not enclosed in square brackets represent their corresponding Web interface and/or EMS parameters.

Note: Some parameters are configurable only through the *ini* file.

6.1 Networking Parameters

This subsection describes the device's networking parameters.

6.1.1 Ethernet Parameters

The Ethernet parameters are described in the table below.

Table 6-1: Ethernet Parameters

Parameter	Description
EMS: Physical Configuration **[EthernetPhyConfiguration]**	Defines the Ethernet connection mode type. ▪ **[0]** = 10Base-T half-duplex ▪ **[1]** = 10Base-T full-duplex ▪ **[2]** = 100Base-TX half-duplex ▪ **[3]** = 100Base-TX full-duplex ▪ **[4]** = Auto-negotiate (default) For detailed information on Ethernet interface configuration, refer to Ethernet Interface Configuration on page 603. **Note:** For this parameter to take effect, a device reset is required.
[MIIRedundancyEnable]	Enables the Ethernet Interface Redundancy feature. When enabled, the device performs a switchover to the second (redundant) Ethernet port upon upon sensing a link failure in the primary Ethernet port. When disabled, the device operates with a single port (i.e. no redundancy support). ▪ **[0]** = Disable ▪ **[1]** = Enable (default) For detailed information on Ethernet interface redundancy, refer to Ethernet Interface Redundancy on page 603. **Note:** For this parameter to take effect, a device reset is required.

6.1.2 Multiple IP Interfaces and VLAN Parameters

The IP network interfaces and VLAN parameters are described in the table below.

Table 6-2: IP Network Interfaces and VLAN Parameters

Parameter	Description
Web: Multiple Interface Table **EMS: IP Interface Settings**	
[InterfaceTable]	This *ini* file table parameter configures the Multiple Interface table for configuring logical IP addresses. The format of this parameter is as follows: [InterfaceTable] FORMAT InterfaceTable_Index = InterfaceTable_ApplicationTypes, InterfaceTable_InterfaceMode, InterfaceTable_IPAddress, InterfaceTable_PrefixLength, InterfaceTable_Gateway, InterfaceTable_VlanID, InterfaceTable_InterfaceName; [\InterfaceTable] For example: InterfaceTable 0 = 0, 0, 192.168.85.14, 16, 0.0.0.0, 1, Management; InterfaceTable 1 = 2, 0, 200.200.85.14, 24, 0.0.0.0, 200, Control; InterfaceTable 2 = 1, 0, 211.211.85.14, 24, 211.211.85.1, 211, Media; The above example, configures three network interfaces (OAMP, Control, and Media). **Notes:** ▪ For this *ini* file table parameter to take effect, a device reset is required. ▪ Up to 16 logical IP addresses with associated VLANs can be defined (indices 0-15). However, only up to 8 interfaces can be used for media RTP traffic (assigned to a Media Realm in the 'SIP Media Realm' table, which in turn is assigned to an IP Group). ▪ Each interface index must be unique. ▪ Each IP interface must have a unique subnet. ▪ Subnets in different interfaces must not be overlapping in any way (e.g., defining two interfaces with 10.0.0.1/8 and 10.50.10.1/24 is invalid). Each interface must have its own address space. ▪ Upon device start up, this table is parsed and passes comprehensive validation tests. If any errors occur during this validation phase, the device sends an error message to the Syslog server and falls back to a “safe mode”, using a single IPv4 interface and without VLANs. Therefore, check the Syslog for any error messages. ▪ When booting using BootP/DHCP protocols, an IP address is obtained from the server. This address is used as the OAMP address for this session, overriding the address configured using the InterfaceTable. The address specified for OAMP applications in this becomes available when booting from flash again. This enables the device to work

Parameter	Description
	with a temporary address for initial management and configuration while retaining the address to be used for deployment. ▪ For configuring additional routing rules for other interfaces, use the 'Outbound IP Routing Table'. ▪ To configure multiple IP interfaces in the Web interface and for a detailed description of the table's parameters, refer to "Configuring the Multiple Interface Table" on page 61). ▪ For a description of configuring *ini* file table parameters, refer to "Configuring *ini* File Table Parameters" on page 222.
Single IP Network Parameters	
Web: IP Address EMS: Local IP Address **[LocalOAMIPAddress]**	The device's source IP address in the operations, administration, maintenance, and provisioning (OAMP) network. The default value is 0.0.0.0. **Note:** For this parameter to take effect, a device reset is required.
Web: Subnet Mask EMS: OAM Subnet Mask **[LocalOAMSubnetMask]**	The device's subnet mask in the OAMP network. The default subnet mask is 0.0.0.0. **Note:** For this parameter to take effect, a device reset is required.
Web: Default Gateway Address EMS: Local Def GW **[LocalOAMDefaultGW]**	N/A. Use the IP Routing table instead.
VLAN Parameters	
Web/EMS: VLAN Mode **[VLANMode]**	Enables the VLAN functionality. ▪ **[0]** Disable (default). ▪ **[1]** Enable = VLAN tagging (IEEE 802.1Q) is enabled. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ VLANs are available only when booting the device from flash. When booting using BootP/DHCP protocols, VLANs are disabled to allow easier maintenance access. In this scenario, multiple network interface capabilities are not available.

Parameter	Description
Web/EMS: Native VLAN ID **[VLANNativeVLANID]**	Defines the VLAN ID to which untagged incoming traffic is assigned. Outgoing packets sent to this VLAN are sent only with a priority tag (VLAN ID = 0). When this parameter is equal to one of the VLAN IDs in the Multiple Interface table (and VLANs are enabled), untagged incoming traffic is considered as incoming traffic for that interface. Outgoing traffic sent from this interface is sent with the priority tag (tagged with VLAN ID = 0). When this parameter is different from any value in the 'VLAN ID' column in the table, untagged incoming traffic is discarded and all outgoing traffic is tagged. **Note:** If this parameter is not set (i.e., default value is 1), but one of the interfaces has a VLAN ID configured to 1, this interface is still considered the 'Native' VLAN. If you do not wish to have a 'Native' VLAN ID and want to use VLAN ID 1, set this parameter to a value other than any VLAN ID in the table.
Web/EMS: OAM VLAN ID **[VLANOamVLANID]**	Defines the OAMP VLAN identifier. The valid range is 1 to 4094. The default value is 1.
Web/EMS: Control VLAN ID **[VLANControlVLANID]**	Defines the Control VLAN identifier. The valid range is 1 to 4094. The default value is 2.
Web/EMS: Media VLAN ID **[VLANMediaVLANID]**	Defines the Media VLAN identifier. The valid range is 1 to 4094. The default value is 3.
[EnableDNSasOAM]	This parameter applies to both Multiple IPs and VLAN mechanisms. Multiple IPs: Determines the network type for DNS services. VLANs: Determines the traffic type for DNS services. ▪ **[1]** = OAMP (default) ▪ **[0]** = Control. **Note:** For this parameter to take effect, a device reset is required.
[EnableNTPasOAM]	This parameter applies to both Multiple IPs and VLAN mechanisms. Multiple IPs: Determines the network type for NTP services. VLANs: Determines the traffic type for NTP services. ▪ **[1]** = OAMP (default) ▪ **[0]** = Control. **Note:** For this parameter to take effect, a device reset is required.
[VLANSendNonTaggedOnNative]	Determines whether to send non-tagged packets on the native VLAN. ▪ **[0]** = Sends priority tag packets (default). ▪ **[1]** = Sends regular packets (with no VLAN tag). **Note:** For this parameter to take effect, a device reset is required.

6.1.3 Static Routing Parameters

The static routing parameters are described in the table below.

Table 6-3: Static Routing Parameters

Parameter	Description
Static IP Routing Table Parameters You can define up to 50 static IP routing rules for the device. For example, you can define static routing rules for the OAMP and Control networks, since a default gateway is supported only for the Media traffic network. Before sending an IP packet, the device searches this table for an entry that matches the requested destination host/network. If such an entry is found, the device sends the packet to the indicated router. If no explicit entry is found, the packet is sent to the default gateway (configured in the 'Multiple Interface' table). The IP routing parameters are array parameters. Each parameter configures a specific column in the IP Routing table. The first entry in each parameter refers to the first row in the IP Routing table, the second entry to the second row, and so on. In the following example, two rows are configured when the device is in network 10.31.x.x: RoutingTableDestinationsColumn = 130.33.4.6, 83.4.87.6 RoutingTableDestinationMasksColumn = 255.255.255.255, 255.255.255.0 RoutingTableGatewaysColumn = 10.31.0.1, 10.31.0.112 RoutingTableInterfacesColumn = 0, 1 RoutingTableHopsCountColumn = 20, 20	
Web: Destination IP Address EMS: Destination IP **[RoutingTableDestinationsColumn]**	Specifies the IP address of the destination host/network. **Note:** For this parameter to take effect, a device reset is required.
Web: Destination Mask EMS: Prefix Length **[RoutingTableDestinationMasksColumn]**	Specifies the subnet mask of the destination host/network. **Note:** For this parameter to take effect, a device reset is required.
Web: Gateway IP Address EMS: Next Hop **[RoutingTableGatewaysColumn]**	The IP address of the router (next hop) to which the packets are sent if their destination matches the rules in the adjacent columns. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ The Gateway address must be in the same subnet as configured on the 'Multiple Interface Table' page (refer to "Configuring the Multiple Interface Table" on page 61).
Web: Metric EMS: Primary Routing Metric **[RoutingTableHopsCountColumn]**	The maximum number of times a packet can be forwarded (hops) between the device and destination (typically, up to 20). **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ This parameter must be set to a number greater than 0 for the routing rule to be valid. Routing entries with Hop Count equals 0 are local routes set automatically by the device.

Parameter	Description
Web: Interface EMS: Interface Index **[RoutingTableInterfacesColumn]**	Specifies the interface (network type) to which the routing rule is applied. ▪ **[0]** = OAMP (default). ▪ **[1]** = Media. ▪ **[2]** = Control. For detailed information on the network types, refer to "Configuring the Multiple Interface Table" on page 61. **Note:** For this parameter to take effect, a device reset is required.

6.1.4 Quality of Service Parameters

The Quality of Service (QoS) parameters are described in the table below. The device allows you to specify values for Layer-2 and Layer-3 priorities by assigning values to the following service classes:

- Network Service class – network control traffic (ICMP, ARP)
- Premium Media service class – used for RTP Media traffic
- Premium Control Service class – used for Call Control traffic
- Gold Service class – used for streaming applications
- Bronze Service class – used for OAMP applications

The Layer-2 QoS parameters enables setting the values for the 3 priority bits in the VLAN tag of frames related to a specific service class (according to the IEEE 802.1p standard). The Layer-3 QoS parameters enables setting the values of the DiffServ field in the IP Header of the frames related to a specific service class.

Table 6-4: QoS Parameters

Parameter	Description
Layer-2 Class Of Service Parameters (VLAN Tag Priority Field)	
Web: Network Priority EMS: Network Service Class Priority **[VLANNetworkServiceClassPriority]**	Defines the VLAN priority (IEEE 802.1p) for Network Class of Service (CoS) content. The valid range is 0 to 7. The default value is 7.
Web: Media Premium EMS: Premium Service Class Media Priority Priority **[VLANPremiumServiceClassMediaPriority]**	Defines the VLAN priority (IEEE 802.1p) for the Premium CoS content and media traffic. The valid range is 0 to 7. The default value is 6.
Web: Control Premium Priority EMS: Premium Service Class Control Priority **[VLANPremiumServiceClassControlPriority]**	Defines the VLAN priority (IEEE 802.1p) for the Premium CoS content and control traffic. The valid range is 0 to 7. The default value is 6.
Web: Gold Priority EMS: Gold Service Class Priority **[VlanGoldServiceClassPriority]**	Defines the VLAN priority (IEEE 802.1p) for the Gold CoS content. The valid range is 0 to 7. The default value is 4.

Parameter	Description
Web: Bronze Priority EMS: Bronze Service Class Priority **[VLANBronzeServiceClassPriority]**	Defines the VLAN priority (IEEE 802.1p) for the Bronze CoS content. The valid range is 0 to 7. The default value is 2.
Layer-3 Class of Service (TOS/DiffServ) Parameters For detailed information on IP QoS via Differentiated Services, refer to "IP QoS via Differentiated Services (DiffServ)" on page 608.	
Web: Network QoS EMS: Network Service Class Diff Serv **[NetworkServiceClassDiffServ]**	Defines the Differentiated Services (DiffServ) value for Network CoS content. The valid range is 0 to 63. The default value is 48.
Web: Media Premium QoS EMS: Premium Service Class Media Diff Serv **[PremiumServiceClassMediaDiffServ]**	Defines the DiffServ value for Premium Media CoS content (only if IPDiffServ is not set in the selected IP Profile). The valid range is 0 to 63. The default value is 46. **Note:** The value for the Premium Control DiffServ is determined by the following (according to priority): ▪ IPDiffServ value in the selected IP Profile. ▪ PremiumServiceClassMediaDiffServ.
Web: Control Premium QoS EMS: Premium Service Class Control Diff Serv **[PremiumServiceClassControlDiffServ]**	Defines the DiffServ value for Premium Control CoS content (only if ControlIPDiffserv is not set in the selected IP Profile). The valid range is 0 to 63. The default value is 40. **Note:** ▪ The value for the Premium Control DiffServ is determined by the following (according to priority): ✓ ControlIPDiffserv value in the selected IP Profile. ✓ PremiumServiceClassControlDiffServ.
Web: Gold QoS EMS: Gold Service Class Diff Serv **[GoldServiceClassDiffServ]**	Defines the DiffServ value for the Gold CoS content. The valid range is 0 to 63. The default value is 26.
Web: Bronze QoS EMS: Bronze Service Class Diff Serv **[BronzeServiceClassDiffServ]**	Defines the DiffServ value for the Bronze CoS content. The valid range is 0 to 63. The default value is 10.

6.1.5 NAT and STUN Parameters

The Network Address Translation (NAT) and Simple Traversal of UDP through NAT (STUN) parameters are described in the table below.

Table 6-5: NAT and STUN Parameters

Parameter	Description
STUN Parameters	
Web: Enable STUN EMS: STUN Enable **[EnableSTUN]**	Determines whether Simple Traversal of UDP through NATs (STUN) is enabled. ▪ **[0]** Disable (default) ▪ **[1]** Enable When enabled, the device functions as a STUN client and communicates with a STUN server located in the public Internet. STUN is used to discover whether the device is located behind a NAT and the type of NAT. In addition, it is used to determine the IP addresses and port numbers that the NAT assigns to outgoing signaling messages (using SIP) and media streams (using RTP, RTCP and T.38). STUN works with many existing NAT types and does not require any special behavior from them. For detailed information on STUN, refer to STUN on page 605. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ For defining the STUN server domain name, use the parameter STUNServerDomainName.
Web: STUN Server Primary IP EMS: Primary Server IP **[STUNServerPrimaryIP]**	Defines the IP address of the primary STUN server. The valid range is the legal IP addresses. The default value is 0.0.0.0. **Note:** For this parameter to take effect, a device reset is required.
Web: STUN Server Secondary IP EMS: Secondary Server IP **[STUNServerSecondaryIP]**	Defines the IP address of the secondary STUN server. The valid range is the legal IP addresses. The default value is 0.0.0.0. **Note:** For this parameter to take effect, a device reset is required.
[STUNServerDomainName]	Defines the domain name for the Simple Traversal of User Datagram Protocol (STUN) server's address (used for retrieving all STUN servers with an SRV query). The STUN client can perform the required SRV query to resolve this domain name to an IP address and port, sort the server list, and use the servers according to the sorted list. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ Use either the STUNServerPrimaryIP or the STUNServerDomainName parameter, with priority to the first one.
NAT Parameters	
EMS: Binding Life Time **[NATBindingDefaultTimeout]**	Defines the default NAT binding lifetime in seconds. STUN refreshes the binding information after this time expires. The valid range is 0 to 2,592,000. The default value is 30.

Parameter	Description
	Note: For this parameter to take effect, a device reset is required.
Web: NAT IP Address EMS: Static NAT IP Address **[StaticNatIP]**	Global (public) IP address of the device to enable static NAT between the device and the Internet. **Note:** For this parameter to take effect, a device reset is required.
EMS: Disable NAT **[DisableNAT]**	Enables or disables the NAT mechanism. ▪ **[0]** = Enabled. ▪ **[1]** = Disabled (default). **Note:** The compare operation that is performed on the IP address is enabled by default and is configured by the parameter EnableIPAddrTranslation. The compare operation that is performed on the UDP port is disabled by default and is configured by the parameter EnableUDPPortTranslation.
[EnableIPAddrTranslation]	Enables IP address translation for RTP, RTCP, and T.38 packets. ▪ **[0]** = Disable IP address translation. ▪ **[1]** = Enable IP address translation (default). ▪ **[2]** = Enable IP address translation for RTP Multiplexing (ThroughPacket™). ▪ **[3]** = Enable IP address translation for all protocols (RTP, RTCP, T.38 and RTP Multiplexing). When enabled, the device compares the source IP address of the first incoming packet to the remote IP address stated in the opening of the channel. If the two IP addresses don't match, the NAT mechanism is activated. Consequently, the remote IP address of the outgoing stream is replaced by the source IP address of the first incoming packet. **Notes:** ▪ The NAT mechanism must be enabled for this parameter to take effect (i.e., the parameter DisableNAT is set to 0). ▪ For information on RTP Multiplexing, refer to RTP Multiplexing (ThroughPacket) on page 601.
[EnableUDPPortTranslation]	▪ **[0]** = Disable UDP port translation (default). ▪ **[1]** = Enable UDP port translation. When enabled, the device compares the source UDP port of the first incoming packet to the remote UDP port stated in the opening of the channel. If the two UDP ports don't match, the NAT mechanism is activated. Consequently, the remote UDP port of the outgoing stream is replaced by the source UDP port of the first incoming packet. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ The NAT mechanism and the IP address translation must be enabled for this parameter to take effect (i.e., set the parameter DisableNAT to 0 and the parameter EnableIpAddrTranslation to 1).

6.1.6 NFS Parameters

The Network File Systems (NFS) configuration parameters are described in the table below.

Table 6-6: NFS Parameters

Parameter	Description
[NFSBasePort]	Start of the range of numbers used for local UDP ports used by the NFS client. The maximum number of local ports is maximum channels plus maximum NFS servers. The valid range is 0 to 65535. The default is 47000.
Web: NFS Table **EMS: NFS Settings**	
[NFSServers]	This *ini* file table parameter defines up to 16 NFS file systems so that the device can access a remote server's shared files and directories for loading cmp, *ini*, and auxiliary files (using the Automatic Update mechanism). As a file system, the NFS is independent of machine types, OSs, and network architectures. Note that an NFS file server can share multiple file systems. There must be a separate row for each remote file system shared by the NFS file server that needs to be accessed by the device. The format of this *ini* file table parameter is as follows: [NFSServers] FORMAT NFSServers_Index = NFSServers_HostOrIP, NFSServers_RootPath, NFSServers_NfsVersion, NFSServers_AuthType, NFSServers_UID, NFSServers_GID, NFSServers_VlanType; [\NFSServers] For example: NFSServers 1 = 101.1.13, /audio1, 3, 1, 0, 1, 1; **Notes:** ▪ You can configure up to 16 NFS file systems (where the first index is 0). ▪ To avoid terminating current calls, a row must not be deleted or modified while the device is currently accessing files on the remote NFS file system. ▪ The combination of Host/IP and Root Path must be unique for each index in the table. For example, the table must include only one index entry with a Host/IP of '192.168.1.1' and Root Path of '/audio'. ▪ This parameter is applicable only if VLANs are enabled or Multiple IPs is configured. ▪ For a detailed description of the table's parameters and to configure NFS using the Web interface, refer to "Configuring the NFS Settings" on page 67. ▪ For a description of configuring *ini* file table parameters, refer to "Configuring *ini* File Table Parameters" on page 222.

6.1.7 DNS Parameters

The Domain name System (DNS) parameters are described in the table below.

Table 6-7: DNS Parameters

Parameter	Description
Web: DNS Primary Server IP EMS: DNS Primary Server **[DNSPriServerIP]**	The IP address of the primary DNS server. Enter the IP address in dotted-decimal notation, for example, 10.8.2.255. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ To use Fully Qualified Domain Names (FQDN) in the 'Outbound IP Routing Table', you must define this parameter.
Web: DNS Secondary Server IP EMS: DNS Secondary Server **[DNSSecServerIP]**	The IP address of the second DNS server. Enter the IP address in dotted-decimal notation, for example, 10.8.2.255. **Note:** For this parameter to take effect, a device reset is required.
Web: Internal DNS Table **EMS: DNS Information**	
[DNS2IP]	This *ini* file table parameter configures the internal DNS table for resolving host names into IP addresses. Up to four different IP addresses (in dotted-decimal notation) can be assigned to a host name. The format of this parameter is as follows: **[Dns2Ip]** FORMAT Dns2Ip_Index = Dns2Ip_DomainName, Dns2Ip_FirstIpAddress, Dns2Ip_SecondIpAddress, Dns2Ip_ThirdIpAddress, Dns2Ip_FourthIpAddress; **[\Dns2Ip]** For example: Dns2Ip 0 = DnsName, 1.1.1.1, 2.2.2.2, 3.3.3.3, 4.4.4.4; **Notes:** ▪ This parameter can include up to 20 indices. ▪ If the internal DNS table is used, the device first attempts to resolve a domain name using this table. If the domain name isn't found, the device performs a DNS resolution using an external DNS server. ▪ To configure the internal DNS table using the Web interface and for a description of the parameters in this *ini* file table parameter, refer to "Configuring the Internal DNS Table" on page 167. ▪ For an explanation on using *ini* file table parameters, refer to "Configuring *ini* File Table Parameters" on page 222.
Web: Internal SRV Table **EMS: DNS Information**	
[SRV2IP]	This *ini* file table parameter defines the internal SRV table for resolving host names into DNS A-Records. Three different A-Records can be assigned to a host name. Each A-Record contains the host name, priority, weight, and port. The format of this parameter is as follows: [SRV2IP]

Parameter	Description
	FORMAT SRV2IP_Index = SRV2IP_InternalDomain, SRV2IP_TransportType, SRV2IP_Dns1, SRV2IP_Priority1, SRV2IP_Weight1, SRV2IP_Port1, SRV2IP_Dns2, SRV2IP_Priority2, SRV2IP_Weight2, SRV2IP_Port2, SRV2IP_Dns3, SRV2IP_Priority3, SRV2IP_Weight3, SRV2IP_Port3; [\SRV2IP] For example: SRV2IP 0 = SrvDomain,0,Dnsname1,1,1,500,Dnsname2,2,2,501,$$,0,0,0; **Notes**: ▪ This parameter can include up to 10 indices. ▪ If the Internal SRV table is used, the device first attempts to resolve a domain name using this table. If the domain name isn't located, the device performs an SRV resolution using an external DNS server. ▪ To configure the Internal SRV table using the Web interface and for a description of the parameters in this *ini* file table parameter, refer to "Configuring the Internal SRV Table" on page 168. ▪ For an explanation on using *ini* file table parameters, refer to "Configuring *ini* File Table Parameters" on page 222.

6.1.8 DHCP Parameters

The Dynamic Host Control Protocol (DHCP) parameters are described in the table below.

Table 6-8: DHCP Parameters

Parameter	Description
Web: Enable DHCP EMS: DHCP Enable **[DHCPEnable]**	Determines whether Dynamic Host Control Protocol (DHCP) is enabled. ▪ **[0]** Disable = Disable DHCP support on the device (default). ▪ **[1]** Enable = Enable DHCP support on the device. After the device powers up, it attempts to communicate with a BootP server. If a BootP server does not respond and DHCP is enabled, then the device attempts to obtain its IP address and other networking parameters from the DHCP server. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ After you enable the DHCP server, perform the following procedure: **a.** Enable DHCP and save the configuration. **b.** Perform a cold reset using the device's hardware reset button (soft reset using the Web interface doesn't trigger the BootP/DHCP procedure and this parameter reverts to 'Disable'). ▪ Throughout the DHCP procedure, the BootP/TFTP application must be deactivated, otherwise the device receives a response from the BootP server instead of from the DHCP server. ▪ For additional information on DHCP, refer to the *Product Reference Manual*.

Parameter	Description
	▪ This parameter is a special 'Hidden' parameter. Once defined and saved in flash memory, its assigned value doesn't revert to its default even if the parameter doesn't appear in the *ini* file.
EMS: DHCP Speed Factor **[DHCPSpeedFactor]**	Determines the DHCP renewal speed. ▪ **[0]** = Disable ▪ **[1]** = Normal (default) ▪ **[2]** to **[10]** = Fast When set to 0, the DHCP lease renewal is disabled. Otherwise, the renewal time is divided by this factor. Some DHCP-enabled routers perform better when set to 4. **Note:** For this parameter to take effect, a device reset is required.
Web: Enable DHCP Lease Renewal **[EnableDHCPLeaseRenewal]**	Enables or disables DHCP renewal support. ▪ **[0]** Disable (default) ▪ **[1]** Enable This parameter is applicable only if the parameter DHCPEnable is set to 0 for cases where booting up the device using DHCP is not desirable but renewing DHCP leasing is. When the device is powered up, it attempts to communicate with a BootP server. If there is no response and if DHCP is disabled, the device boots from flash. It then attempts to communicate with the DHCP server to renew the lease. **Note:** For this parameter to take effect, a device reset is required.

6.1.9 NTP and Daylight Saving Time Parameters

The Network Time Protocol (NTP) and daylight saving time parameters are described in the table below.

Table 6-9: NTP and Daylight Saving Time Parameters

Parameter	Description
NTP Parameters **Note:** For detailed information on Network Time Protocol (NTP), refer to "Simple Network Time Protocol Support" on page 607.	
Web: NTP Server IP Address EMS: Server IP Address **[NTPServerIP]**	The IP address (in dotted-decimal notation) of the NTP server. The default IP address is 0.0.0.0 (i.e., internal NTP client is disabled).
Web: NTP UTC Offset EMS: UTC Offset **[NTPServerUTCOffset]**	Defines the Universal Time Coordinate (UTC) offset (in seconds) from the NTP server. The default offset is 0. The offset range is -43200 to 43200.
Web: NTP Update Interval EMS: Update Interval **[NTPUpdateInterval]**	Defines the time interval (in seconds) that the NTP client requests for a time update. The default interval is 86400 (i.e., 24 hours). The range is 0 to 214783647. **Note:** It is not recommend to set this parameter to beyond one month (i.e., 2592000 seconds).

Parameter	Description
Daylight Saving Time Parameters	
Web: Day Light Saving Time EMS: Mode **[DayLightSavingTimeEnable]**	Determines whether to enable daylight saving time. ▪ **[0]** Disable (default) ▪ **[1]** Enable
Web: Start Time EMS: Start **[DayLightSavingTimeStart]**	Defines the date and time when daylight saving begins. The format of the value is mo:dd:hh:mm (month, day, hour, and minutes).
Web: End Time EMS: End **[DayLightSavingTimeEnd]**	Defines the date and time when daylight saving ends. The format of the value is mo:dd:hh:mm (month, day, hour, and minutes).
Web/EMS: Offset **[DayLightSavingTimeOffset]**	Daylight saving time offset (in minutes). The valid range is 0 to 120. The default is 60.

6.2 Web and Telnet Parameters

This subsection describes the device's Web and Telnet parameters.

6.2.1 General Parameters

The general Web and Telnet parameters are described in the table below.

Table 6-10: General Web and Telnet Parameters

Parameter	Description
Web: Web and Telnet Access List Table EMS: Web Access Addresses **[WebAccessList_x]**	Defines up to ten IP addresses that are permitted to access the device's Web interface and Telnet interfaces. Access from an undefined IP address is denied. When no IP addresses are defined in this table, this security feature is inactive (i.e., the device can be accessed from any IP address). The default value is 0.0.0.0 (i.e., the device can be accessed from any IP address). For example: WebAccessList_0 = 10.13.2.66 WebAccessList_1 = 10.13.77.7 For defining the Web and Telnet Access list using the Web interface, refer to "Configuring the Web and Telnet Access List" on page 85.
Web: Use RADIUS for Web/Telnet Login EMS: Web Use Radius Login **[WebRADIUSLogin]**	Uses RADIUS queries for Web and Telnet interface authentication. ▪ **[0]** Disable (default). ▪ **[1]** Enable. When enabled, logging in to the device's Web and Telnet embedded servers is performed through a RADIUS server. The device contacts a user-defined server and verifies the given user name and password pair against a remote database, in a secure manner. **Notes:** ▪ The parameter EnableRADIUS must be set to 1. ▪ RADIUS authentication requires HTTP basic authentication, meaning

Parameter	Description
	the user name and password are transmitted in clear text over the network. Therefore, it's recommended to set the parameter HTTPSOnly to 1 to force the use of HTTPS, since the transport is encrypted. ▪ If using RADIUS authentication when logging in to the CLI, only the primary Web User Account (which has Security Administration access level) can access the device's CLI (refer to "Configuring the Web User Accounts" on page 83).

6.2.2 Web Parameters

The Web parameters are described in the table below.

Table 6-11: Web Parameters

Parameter	Description
[DisableWebTask]	Disables or enables device management through the Web interface. ▪ **[0]** = Enable Web management (default). ▪ **[1]** = Disable Web management. **Note:** For this parameter to take effect, a device reset is required.
[HTTPport]	HTTP port used for Web management (default is 80). **Note:** For this parameter to take effect, a device reset is required.
EMS: Disable WEB Config **[DisableWebConfig]**	Determines whether the entire Web interface is in read-only mode. ▪ **[0]** = Enables modifications of parameters (default). ▪ **[1]** = Web interface in read-only mode. When in read-only mode, parameters can't be modified. In addition, the following pages can't be accessed: 'Web User Accounts', 'Certificates', 'Regional Settings', 'Maintenance Actions' and all file-loading pages ('Load Auxiliary Files', 'Software Upgrade Wizard', and 'Configuration File'). **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ To return to read/write after you have applied read-only using this parameter (set to 1), you need to reboot your device with an *ini* file that doesn't include this parameter, using the BootP/TFTP Server utility (refer to the Product Reference Manual).
[ResetWebPassword]	Resets the username and password of the primary and secondary accounts to their defaults. ▪ **[0]** = Password and username retain their values (default). ▪ **[1]** = Password and username are reset (for the default username and password, refer to User Accounts). **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ The username and password cannot be reset from the Web interface (i.e., via AdminPage or by loading an *ini* file).

Parameter	Description
[ScenarioFileName]	Defines the file name of the Scenario file to be loaded to the device. The file name must have the *.dat extension and can be up to 47 characters. For loading a Scenario using the Web interface, refer to Loading a Scenario to the Device on page 48.
[WelcomeMessage]	This *ini* file table parameter configures the Welcome message that appears after a Web interface login. The format of this parameter is as follows: [WelcomeMessage] FORMAT WelcomeMessage_Index = WelcomeMessage_Text WelcomeMessage 1 = "..." ; WelcomeMessage 2 = "..." ; WelcomeMessage 3 = "..." ; [\WelcomeMessage] For Example: [WelcomeMessage] FORMAT WelcomeMessage_Index = WelcomeMessage_Text WelcomeMessage 1 = "**********" ; WelcomeMessage 2 = "********* This is a Welcome message ***" ; WelcomeMessage 3 = "**********" ; [\WelcomeMessage] **Notes:** ▪ Each index represents a line of text in the Welcome message box. Up to 20 indices can be defined. ▪ If this parameter is not configured, no Welcome message is displayed. ▪ For a description on using *ini* file table parameters, refer to "Configuring *ini* File Table Parameters" on page 222.

6.2.3 Telnet Parameters

The Telnet parameters are described in the table below.

Table 6-12: Telnet Parameters

Parameter	Description
Web: Embedded Telnet Server EMS: Server Enable **[TelnetServerEnable]**	Enables or disables the device's embedded Telnet server. Telnet is disabled by default for security. ▪ **[0]** Disable (default) ▪ **[1]** Enable Unsecured ▪ **[2]** Enable Secured (SSL) **Note:** Only the primary Web User Account (which has Security Administration access level) can access the device using Telnet (refer to "Configuring the Web User Accounts" on page 83).
Web: Telnet Server TCP Port EMS: Server Port **[TelnetServerPort]**	Defines the port number for the embedded Telnet server. The valid range is all valid port numbers. The default port is 23.

Parameter	Description
Web: Telnet Server Idle Timeout EMS: Server Idle Disconnect **[TelnetServerIdleDisconnect]**	Defines the timeout (in minutes) for disconnection of an idle Telnet session. When set to zero, idle sessions are not disconnected. The valid range is any value. The default value is 0. **Note:** For this parameter to take effect, a device reset is required.

6.3 Debugging and Diagnostics Parameters

This subsection describes the device's debugging and diagnostic parameters.

6.3.1 General Parameters

The general debugging and diagnostic parameters are described in the table below.

Table 6-13: General Debugging and Diagnostic Parameters

Parameter	Description
EMS: Enable Diagnostics **[EnableDiagnostics]**	Checks the correct functionality of the different hardware components on the device. On completion of the check and if the test fails, the device sends information on the test results of each hardware component to the Syslog server. ▪ **[0]** = Rapid and Enhanced self-test mode (default). ▪ **[1]** = Detailed self-test mode (full test of DSPs, PCM, Switch, LAN, PHY and Flash). ▪ **[2]** = A quicker version of the Detailed self-test mode (full test of DSPs, PCM, Switch, LAN, PHY, but partial test of Flash). For detailed information, refer to the *Product Reference Manual*. **Note:** For this parameter to take effect, a device reset is required.
Web: Enable LAN Watchdog **[EnableLanWatchDog]**	Determines whether the LAN Watch-Dog feature is enabled**.** ▪ **[0]** Disable = Disable LAN Watch-Dog (default). ▪ **[1]** Enable = Enable LAN Watch-Dog. When LAN Watch-Dog is enabled, the device's overall communication integrity is checked periodically. If no communication is detected for about three minutes, the device performs a self test: ▪ If the self-test succeeds, the problem is a logical link down (i.e., Ethernet cable disconnected on the switch side) and the Busy Out mechanism is activated if enabled (i.e., the parameter EnableBusyOut is set to 1). ▪ If the self-test fails, the device restarts to overcome internal fatal communication error. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ Enable LAN Watchdog is relevant only if the Ethernet connection is full duplex.

Parameter	Description
[WatchDogStatus]	▪ **[0]** = Disable device's watch dog. ▪ **[1]** = Enable device's watch dog (default). **Note:** For this parameter to take effect, a device reset is required.
[LifeLineType]	Defines the scenario upon which the Lifeline phone is activated. The Lifeline phone is available on Port 1 of each FXS module.FXS Port 1 is connected to the POTS (Lifeline) phone as well as to the PSTN/PBX (using a splitter cable). Upon power outage and/or network failure, PSTN connectivity is maintained for the FXS phone user. ▪ **[0]** = Lifeline is activated upon power failure (default). ▪ **[1]** = Lifeline is activated upon power failure or when the link is down (physically disconnected). ▪ **[2]** = Lifeline is activated upon power failure, when the link is down, or upon network failure (logical link disconnected). **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ This parameter is applicable only to FXS interfaces. ▪ To enable Lifeline switching on network failure, the LAN watch dog must be activated (i.e., set the parameter EnableLANWatchDog to 1). ▪ For a detailed description on cabling the device for Lifeline, refer to the device's Installation Manual.
Web: Delay After Reset **[sec]** **[GWAppDelayTime]**	Defines the time interval (in seconds) that the device's operation is delayed after a reset. The valid range is 0 to 45. The default value is 7 seconds. **Note:** This feature helps overcome connection problems caused by some LAN routers or IP configuration parameters' modifications by a DHCP server.
[Mediant1000DualPowerSupplySupported]	Determines whether the device sends raised alarms (to the SNMP client and/or Web interface) concerned with the Power Supply modules. ▪ **[1]** (default) = No alarms are sent. ▪ **[2]** = The device sends alarms if one of the Power Supply modules is removed from the chassis. These alarms are reflected in the SNMP and Web interface. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ If this parameter is set to 2 and for this feature to be functional, both Main and Redundant Power Supply modules must be present in the chassis.
[GroundKeyDetection]	Enables analog ground-key detection for the device. The device's FXS and FXO modules implement ground-start signaling. When disabled, the device uses loop-start signaling. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable (enables ground start) **Notes:** ▪ For this parameter to take effect, a device reset is required.

Parameter	Description
	▪ For ground-start signaling, ensure that the FXO G module is installed (and not the regular FXO module) in the device's chassis. ▪ For FXO ground-start signaling, ensure that the parameter EnableCurrentDisconnect is set to 1 and the parameter FXOBetweenRingTime is set to 300. ▪ FXS ground-start interface does not generate a ringing voltage. The FXS interface initiates the signaling by grounding of the TIP lead.

6.3.2 Syslog, CDR and Debug Parameters

The Syslog, CDR and debug parameters are described in the table below.

Table 6-14: Syslog, CDR and Debug Parameters

Parameter	Description
Web/EMS: Syslog Server IP Address **[SyslogServerIP]**	IP address (in dotted-decimal notation) of the computer you are using to run the Syslog server. The Syslog server is an application designed to collect the logs and error messages generated by the device. Default IP address is 0.0.0.0. For information on Syslog, refer to the *Product Reference Manual*.
Web: Syslog Server Port EMS: Syslog Server Port Number **[SyslogServerPort]**	Defines the UDP port of the Syslog server. The valid range is 0 to 65,535. The default port is 514. For information on the Syslog, refer to the *Product Reference Manual*.

Parameter	Description
Web: Enable Syslog EMS: Syslog enable **[EnableSyslog]**	Sends the logs and error message generated by the device to the Syslog server. ▪ **[0]** Disable = Logs and errors are not sent to the Syslog server (default). ▪ **[1]** Enable = Enables the Syslog server. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ If you enable Syslog, you must enter an IP address and a port number (using the SyslogServerIP and SyslogServerPort parameters). ▪ You can configure the device to send Syslog messages implementing Debug Recording, by using the SyslogOutputMethod parameter. For a detailed description on Debug Recording, refer to the *Product Reference Manual*. ▪ Syslog messages may increase the network traffic. ▪ To configure Syslog logging levels, use the parameter GwDebugLevel. ▪ For information on the Syslog, refer to the *Product Reference Manual*. ▪ Logs are also sent to the RS-232 serial port. For information on establishing a serial communications link with the device, refer to the device's Installation Manual.
[SyslogOutputMethod]	Determines the method used for Syslog messages. ▪ **[0]** = Send all Syslog messages to the defined Syslog server (default). ▪ **[1]** = Send all Syslog messages using the Debug Recording mechanism. ▪ **[2]** = Send only Error and Warning level Syslog messages using the Debug Recording mechanism. For a detailed description on Debug Recording, refer to the *Product Reference Manual*.
[MaxBundleSyslogLength]	The maximum size (in bytes) threshold of logged Syslog messages bundled into a single UDP packet, after which they are sent to a Syslog server. The valid value range is 0 to 1220 (where 0 indicates that no bundling occurs). The default is 1220. **Note:** This parameter is applicable only if the GWDebugLevel parameter is set to 7.
Web: CDR Server IP Address EMS: IP Address of CDR Server **[CDRSyslogServerIP]**	Defines the destination IP address to where CDR logs are sent. The default value is a null string, which causes CDR messages to be sent with all Syslog messages to the Syslog server. **Notes:** ▪ The CDR messages are sent to UDP port 514 (default Syslog port). ▪ This mechanism is active only when Syslog is enabled (i.e., the parameter EnableSyslog is set to 1).

Parameter	Description
Web/EMS: CDR Report Level **[CDRReportLevel]**	Determines whether Call Detail Records (CDR) are sent to the Syslog server and when they are sent. ▪ **[0]** None = CDRs are not used (default). ▪ **[1]** End Call = CDR is sent to the Syslog server at the end of each call. ▪ **[2]** Start & End Call = CDR report is sent to Syslog at the start and end of each call. ▪ **[3]** Connect & End Call = CDR report is sent to Syslog at connection and at the end of each call. ▪ **[4]** Start & Connect & End Call = CDR report is sent to Syslog at the start, at connection, and at the end of each call. **Notes:** ▪ The CDR Syslog message complies with RFC 3161 and is identified by: Facility = 17 (local1) and Severity = 6 (Informational). ▪ This mechanism is active only when Syslog is enabled (i.e., the parameter EnableSyslog is set to 1).
Web/EMS: Debug Level **[GwDebugLevel]**	Syslog debug logging level. ▪ **[0]** 0 (default) = Debug is disabled. ▪ **[1]** 1 = Flow debugging is enabled. ▪ **[5]** 5 = Flow, device interface, stack interface, session manager, and device interface expanded debugging are enabled. ▪ **[7]** 7 = The Syslog debug level automatically changes between level 5, level 1, and level 0, depending on the device's CPU consumption. **Notes:** ▪ Usually set to 5 if debug traces are required. ▪ Options 2, 3, 4, and 6 are not recommended for use.
Web: Activity Types to Report via Activity Log Messages **[ActivityListToLog]**	The Activity Log mechanism enables the device to send log messages (to a Syslog server) for reporting certain types of Web operations according to the below user-defined filters. ▪ **[PVC]** Parameters Value Change = Changes made on-the-fly to parameters. ▪ **[AFL]** Auxiliary Files Loading = Loading of auxiliary files. ▪ **[DR]** Device Reset = Reset of device via the 'Maintenance Actions' page. ▪ **[FB]** Flash Memory Burning = Burning of files or parameters to flash (in 'Maintenance Actions' page). ▪ **[SWU]** Device Software Update = cmp file loading via the Software Upgrade Wizard. ▪ **[ARD]** Access to Restricted Domains = Access to restricted domains, which include the following Web pages: ✓ (1) *ini* parameters (AdminPage) ✓ (2) 'General Security Settings' ✓ (3) 'Configuration File' ✓ (4) 'IPSec/IKE' tables ✓ (5) 'Software Upgrade Key' ✓ (6) 'Internal Firewall'

Parameter	Description
	✓ (7) 'Web Access List' ✓ (8) 'Web User Accounts' ▪ **[NAA]** Non Authorized Access = Attempt to access the Web interface with a false or empty user name or password. ▪ **[SPC]** Sensitive Parameters Value Change = Changes made to sensitive parameters: ✓ (1) IP Address ✓ (2) Subnet Mask ✓ (3) Default Gateway IP Address ✓ (4) ActivityListToLog For example: ActivityListToLog = 'pvc', 'afl', 'dr', 'fb', 'swu', 'ard', 'naa', 'spc'

6.3.3 Remote Alarm Indication Parameters

The Remote Alarm Indication (RAI) parameters are described in the table below.

Table 6-15: RAI Parameters

Parameter	Description
[EnableRAI]	Enables RAI alarm generation if the device's busy endpoints exceed a user-defined threshold. ▪ **[0]** = Disable RAI (Resource Available Indication) service (default). ▪ **[1]** = RAI service enabled and an SNMP 'acBoardCallResourcesAlarm' Alarm Trap is sent.
[RAIHighThreshold]	High threshold percentage of total calls that are active (busy endpoints). When the percentage of the device's busy endpoints exceeds this high threshold, the device sends the SNMP acBoardCallResourcesAlarm alarm trap with a 'major' alarm status. The range is 0 to 100. The default value is 90. **Note:** The percentage of busy endpoints is calculated by dividing the number of busy endpoints by the total number of "enabled" endpoints (trunks are physically connected and synchronized with no alarms and endpoints are defined in the Trunk Group Table).
[RAILowThreshold]	Low threshold percentage of total calls that are active (busy endpoints). When the percentage of the device's busy endpoints falls below this low threshold, the device sends an SNMP acBoardCallResourcesAlarm alarm trap with a 'cleared' alarm status. The range is 0 to 100%. The default value is 90%.
[RAILoopTime]	Time interval (in seconds) that the device periodically checks call resource availability. The valid range is 1 to 200. The default is 10.

6.3.4 Serial Parameters

The RS-232 serial parameters are described in the table below. (Serial interface is mainly used for debugging and for SMDI.)

Table 6-16: Serial Parameters

Parameter	Description
[DisableRS232]	Enables or disables the device's RS-232 port. ▪ **[0]** = RS-232 serial port is enabled (default). ▪ **[1]** = RS-232 serial port is disabled. The RS-232 serial port can be used to change the networking parameters and view error/notification messages. For information on establishing a serial communications link with the device, refer to the device's *Installation Manual*. **Note:** For this parameter to take effect, a device reset is required.
EMS: Baud Rate **[SerialBaudRate]**	Determines the value of the RS-232 baud rate. The valid values include the following: 1200, 2400, 9600, 14400, 19200, 38400, 57600, or 115200 (default). **Note:** For this parameter to take effect, a device reset is required.
EMS: Data **[SerialData]**	Determines the value of the RS-232 data bit. ▪ **[7]** = 7-bit. ▪ **[8]** = 8-bit (default). **Note:** For this parameter to take effect, a device reset is required.
EMS: Parity **[SerialParity]**	Determines the value of the RS-232 polarity. ▪ **[0]** = None (default). ▪ **[1]** = Odd. ▪ **[2]** = Even. **Note:** For this parameter to take effect, a device reset is required.
EMS: Stop **[SerialStop]**	Determines the value of the RS-232 stop bit. ▪ **[1]** = 1-bit (default). ▪ **[2]** = 2-bit. **Note:** For this parameter to take effect, a device reset is required.
EMS: Flow Control **[SerialFlowControl]**	Determines the value of the RS-232 flow control. ▪ **[0]** = None (default). ▪ **[1]** = Hardware. **Note:** For this parameter to take effect, a device reset is required.

6.3.5 BootP Parameters

The BootP parameters are described in the table below. The BootP parameters are special 'hidden' parameters. Once defined and saved in the device's flash memory, they are used even if they don't appear in the *ini* file.

Table 6-17: BootP Parameters

<table>
<tr><th>Parameter</th><th colspan="2">Description</th></tr>
<tr><td rowspan="2">[BootPRetries]</td><td colspan="2">Note: For this parameter to take effect, a device reset is required.
This parameter is used to:</td></tr>
<tr><td>Sets the number of BootP requests the device sends during start-up. The device stops sending BootP requests when either BootP reply is received or number of retries is reached.
▪ [1] = 1 BootP retry, 1 sec.
▪ [2] = 2 BootP retries, 3 sec.
▪ [3] = 3 BootP retries, 6 sec. (default).
▪ [4] = 10 BootP retries, 30 sec.
▪ [5] = 20 BootP retries, 60 sec.
▪ [6] = 40 BootP retries, 120 sec.
▪ [7] = 100 BootP retries, 300 sec.
▪ [15] = BootP retries indefinitely.</td><td>Sets the number of DHCP packets the device sends. If after all packets are sent there's still no reply, the device loads from flash.
▪ [1] = 4 DHCP packets
▪ [2] = 5 DHCP packets
▪ [3] = 6 DHCP packets (default)
▪ [4] = 7 DHCP packets
▪ [5] = 8 DHCP packets
▪ [6] = 9 DHCP packets
▪ [7] = 10 DHCP packets
▪ [15] = 18 DHCP packets</td></tr>
<tr><td>[BootPSelectiveEnable]</td><td colspan="2">Enables the Selective BootP mechanism.
▪ [1] = Enabled.
▪ [0] = Disabled (default).
The Selective BootP mechanism (available from Boot version 1.92) enables the device's integral BootP client to filter unsolicited BootP/DHCP replies (accepts only BootP replies that contain the text 'AUDC' in the vendor specific information field). This option is useful in environments where enterprise BootP/DHCP servers provide undesired responses to the device's BootP requests.
Notes:
▪ For this parameter to take effect, a device reset is required.
▪ When working with DHCP (i.e., the parameter DHCPEnable is set to 1), the selective BootP feature must be disabled.</td></tr>
<tr><td>[BootPDelay]</td><td colspan="2">The interval between the device's startup and the first BootP/DHCP request that is issued by the device.
▪ [1] = 1 second (default).
▪ [2] = 3 second.
▪ [3] = 6 second.
▪ [4] = 30 second.
▪ [5] = 60 second.
Note: For this parameter to take effect, a device reset is required.</td></tr>
</table>

Parameter	Description
[ExtBootPReqEnable]	▪ **[0]** = Disable (default). ▪ **[1]** = Enable extended information to be sent in BootP request. If enabled, the device uses the Vendor Specific Information field in the BootP request to provide device-related initial startup information such as blade type, current IP address, software version. For a full list of the Vendor Specific Information fields, refer to the *Product Reference Manual*. The BootP/TFTP configuration utility displays this information in the 'Client Info' column. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ This option is not available on DHCP servers.

6.4 Security Parameters

This subsection describes the device's security parameters.

6.4.1 General Parameters

The general security parameters are described in the table below.

Table 6-18: General Security Parameters

Parameter	Description
Web: Voice Menu Password **[VoiceMenuPassword]**	The password for accessing the device's voice menu for configuration and status. To activate the menu, connect a POTS telephone and dial *** (three stars) followed by the password. The default value is 12345. For detailed information on the voice menu, refer to the device's Installation Manual.
[EnableSecureStartup]	Enables the Secure Startup mode. In this mode, downloading the *ini* file to the device is restricted to a URL provided in initial configuration (see the parameter IniFileURL) or using DHCP. ▪ **[0]** Disable (default). ▪ **[1]** Enable = disables TFTP and allows secure protocols such as HTTPS to fetch the device configuration. For a detailed explanation on Secure Startup, refer to the Product Reference Manual. **Note:** For this parameter to take effect, a device reset is required.

Parameter	Description
Web: Internal Firewall Parameters **EMS: Firewall Settings**	
[AccessList]	This *ini* file table parameter configures the device's access list (firewall), which defines network traffic filtering rules. For each packet received on the network interface, the table is scanned from the top down until a matching rule is found. This rule can either deny (block) or permit (allow) the packet. Once a rule in the table is located, subsequent rules further down the table are ignored. If the end of the table is reached without a match, the packet is accepted. The format of this parameter is as follows: [ACCESSLIST] FORMAT AccessList_Index = AccessList_Source_IP, AccessList_PrefixLen, AccessList_Start_Port, AccessList_End_Port, AccessList_Protocol, AccessList_Packet_Size, AccessList_Byte_Rate, AccessList_Byte_Burst, AccessList_Allow_Type; [\ACCESSLIST] For example: AccessList 10 = mgmt.customer.com, 32, 0, 80, tcp, 0, 0, 0, allow; AccessList 22 = 10.4.0.0, 16, 4000, 9000, any, 0, 0, 0, block; In the example above, Rule #10 allows traffic from the host 'mgmt.customer.com' destined to TCP ports 0 to 80. Rule #22 blocks traffic from the subnet 10.4.xxx.yyy destined to ports 4000 to 9000. **Notes:** ▪ This parameter can include up to 50 indices. ▪ To configure the firewall using the Web interface and for a description of the parameters of this *ini* file table parameter, refer to "Configuring the Firewall Settings" on page 86. ▪ For a description of configuring with *ini* file table parameters, refer to Configuring *ini* File Table Parameters on page 222.

6.4.2 HTTPS Parameters

The Secure Hypertext Transport Protocol (HTTPS) parameters are described in the table below.

Table 6-19: HTTPS Parameters

Parameter	Description
Web: Secured Web Connection (HTTPS) EMS: HTTPS Only **[HTTPSOnly]**	Determines the protocol used to access the Web interface. ▪ **[0]** HTTP and HTTPS (default). ▪ **[1]** HTTPs Only = Unencrypted HTTP packets are blocked. **Note:** For this parameter to take effect, a device reset is required.
EMS: HTTPS Port **[HTTPSPort]**	Determines the local Secured HTTPS port of the device. The valid range is 1 to 65535 (other restrictions may apply within this range). The default port is 443. **Note:** For this parameter to take effect, a device reset is required.

Parameter	Description
EMS: HTTPS Cipher String **[HTTPSCipherString]**	Defines the Cipher string for HTTPS (in OpenSSL cipher list format). For the valid range values, refer to URL http://www.openssl.org/docs/apps/ciphers.html. The default value is 'EXP' (Export encryption algorithms). For example, use 'ALL' for all ciphers suites. The only ciphers available are RC4 and DES, and the cipher bit strength is limited to 56 bits. **Notes:** ▪ If the "Strong Encryption" Software Upgrade Key is enabled, the default of the HTTPSCipherString parameter is changed to 'RC4:EXP', enabling RC-128bit encryption. ▪ The value 'ALL' can be configured only if the "Strong Encryption" Software Upgrade Key is enabled.
Web: HTTP Authentication Mode EMS: Web Authentication Mode **[WebAuthMode]**	Determines the authentication mode for the Web interface. ▪ **[0]** Basic Mode = Basic authentication (clear text) is used (default). ▪ **[1]** Digest When Possible = Digest authentication (MD5) is used. ▪ **[2]** Basic if HTTPS, Digest if HTTP = Digest authentication (MD5) is used for HTTP, and basic authentication is used for HTTPS. **Note:** When RADIUS login is enabled (i.e., the parameter WebRADIUSLogin is set to 1), basic authentication is forced.
[HTTPSRequireClientCertificate]	Requires client certificates for HTTPS connection. The client certificate must be preloaded to the device and its matching private key must be installed on the managing PC. Time and date must be correctly set on the device for the client certificate to be verified. ▪ **[0]** = Client certificates are not required (default). ▪ **[1]** = Client certificates are required. **Note:** For this parameter to take effect, a device reset is required.
[HTTPSRootFileName]	Defines the name of the HTTPS trusted root certificate file to be loaded using TFTP. The file must be in base64-encoded PEM (Privacy Enhanced Mail) format. The valid range is a 47-character string. **Note:** This parameter is only applicable when the device is loaded using BootP/TFTP. For information on loading this file using the Web interface, refer to the Product Reference Manual.
[HTTPSPkeyFileName]	Defines the name of a private key file (in unencrypted PEM format) to be loaded from the TFTP server.
[HTTPSCertFileName]	Defines the name of the HTTPS server certificate file to be loaded using TFTP. The file must be in base64-encoded PEM format. The valid range is a 47-character string. **Note:** This parameter is only applicable when the device is loaded using BootP/TFTP. For information on loading this file using the Web interface, refer to the Product Reference Manual.

6.4.3 SRTP Parameters

The Secure Real-Time Transport Protocol (SRTP) parameters are described in the table below.

Table 6-20: SRTP Parameters

Parameter	Description
Web: Media Security EMS: Enable Media Security **[EnableMediaSecurity]**	Enables Secure Real-Time Transport Protocol (SRTP). ▪ **[0]** Disable = SRTP is disabled (default). ▪ **[1]** Enable = SRTP is enabled. **Note:** For this parameter to take effect, a device reset is required.
Web/EMS: Media Security Behavior **[MediaSecurityBehaviour]**	Determines the device's mode of operation when SRTP is used (i.e., when the parameter EnableMediaSecurity is set to 1). ▪ **[0]** Preferable = The device initiates encrypted calls. If negotiation of the cipher suite fails, an unencrypted call is established. Incoming calls that don't include encryption information are accepted. (default) ▪ **[1]** Mandatory = The device initiates encrypted calls, but if negotiation of the cipher suite fails, the call is terminated. Incoming calls that don't include encryption information are rejected. ▪ **[2]** Preferable - Single Media = The device sends SDP with only a single media ('m=') line (e.g., m=audio 6000 **RTP/AVP** 4 0 70 96) with RTP/AVP and crypto keys. If the remote SIP UA does not support SRTP, it ignores the crypto lines. **Note:** Before configuring this parameter, set the parameter EnableMediaSecurity parameter to 1.
Web: Master Key Identifier (MKI) Size EMS: Packet MKI Size **[SRTPTxPacketMKISize]**	Determines the size (in bytes) of the Master Key Identifier (MKI) in SRTP Tx packets. The range is 0 to 4. The default value is 0.
Web/EMS: SRTP offered Suites **[SRTPofferedSuites]**	Defines the offered SRTP crypto suites. ▪ **[0]** All = All available crypto suites (default) ▪ **[1]** AES_CM_128_HMAC_SHA1_80 = device uses AES-CM encryption with a 128-bit key and HMAC-SHA1 message authentication with a 80-bit tag. ▪ **[2]** AES_CM_128_HMAC_SHA1_32 = device uses AES-CM encryption with a 128-bit key and HMAC-SHA1 message authentication with a 32-bit tag.
Web: Disable Authentication On Transmitted RTP Packets EMS: RTP AuthenticationDisable Tx **[RTPAuthenticationDisableTx]**	On a secured RTP session, this parameter determines whether to enable authentication on transmitted RTP packets. ▪ **[0]** Enable (default) ▪ **[1]** Disable
Web: Disable Encryption On Transmitted RTP Packets EMS: RTP EncryptionDisable Tx **[RTPEncryptionDisableTx]**	On a secured RTP session, this parameter determines whether to enable encryption on transmitted RTP packets. ▪ **[0]** Enable (default) ▪ **[1]** Disable

Parameter	Description
Web: Disable Encryption On Transmitted RTCP Packets EMS: RTCP EncryptionDisable Tx **[RTCPEncryptionDisableTx]**	On a secured RTP session, this parameter determines whether to enable encryption on transmitted RTCP packets. ▪ **[0]** Enable (default) ▪ **[1]** Disable

6.4.4 TLS Parameters

The Transport Layer Security (TLS) parameters are described in the table below.

Table 6-21: TLS Parameters

Parameter	Description
Web/EMS: TLS Version **[TLSVersion]**	Defines the supported versions of SSL/TLS (Secure Socket Layer/Transport Layer Security. ▪ **[0]** SSL 2.0-3.0 and TLS 1.0 = SSL 2.0, SSL 3.0, and TLS 1.0 are supported (default). ▪ **[1]** TLS 1.0 Only = only TLS 1.0 is used. When set to 0, SSL/TLS handshakes always start with SSL 2.0 and switch to TLS 1.0 if both peers support it. When set to 1, TLS 1.0 is the only version supported; clients attempting to contact the device using SSL 2.0 are rejected. **Note:** For this parameter to take effect, a device reset is required.
Web: TLS Client Re-Handshake Interval EMS: TLS Re Handshake Interval **[TLSReHandshakeInterval]**	Defines the time interval (in minutes) between TLS Re-Handshakes initiated by the device. The interval range is 0 to 1,500 minutes. The default is 0 (i.e., no TLS Re-Handshake).
Web: TLS Mutual Authentication EMS: SIPS Require Client Certificate **[SIPSRequireClientCertificate]**	Determines the device's behavior when acting as a server for TLS connections. ▪ **[0]** Disable = The device does not request the client certificate (default). ▪ **[1]** Enable = The device requires receipt and verification of the client certificate to establish the TLS connection. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ The SIPS certificate files can be changed using the parameters HTTPSCertFileName and HTTPSRootFileName.
Web/EMS: Peer Host Name Verification Mode **[PeerHostNameVerificationMode]**	Determines whether the device verifies the Subject Name of a remote certificate when establishing TLS connections. ▪ **[0]** Disable = Disable (default). ▪ **[1]** Server Only = Verify Subject Name only when acting as a server for the TLS connection. ▪ **[2]** Server & Client = Verify Subject Name when acting as a server or client for the TLS connection.

Parameter	Description
	When a remote certificate is received and this parameter is not disabled, the value of SubjectAltName is compared with the list of available Proxies. If a match is found for any of the configured Proxies, the TLS connection is established. The comparison is performed if the SubjectAltName is either a DNS name (DNSName) or an IP address. If no match is found and the SubjectAltName is marked as 'critical', the TLS connection is not established. If DNSName is used, the certificate can also use wildcards ('*') to replace parts of the domain name. If the SubjectAltName is not marked as 'critical' and there is no match, the CN value of the SubjectName field is compared with the parameter TLSRemoteSubjectName. If a match is found, the connection is established. Otherwise, the connection is terminated.
Web: TLS Client Verify Server Certificate EMS: Verify Server Certificate **[VerifyServerCertificate]**	Determines whether the device, when acting as a client for TLS connections, verifies the Server certificate. The certificate is verified with the Root CA information. ▪ **[0]** Disable (default). ▪ **[1]** Enable. **Note:** If Subject Name verification is necessary, the parameter PeerHostNameVerificationMode must be used as well.
Web/EMS: TLS Remote Subject Name **[TLSRemoteSubjectName]**	Defines the Subject Name that is compared with the name defined in the remote side certificate when establishing TLS connections. If the SubjectAltName of the received certificate is not equal to any of the defined Proxies Host names/IP addresses and is not marked as 'critical', the Common Name (CN) of the Subject field is compared with this value. If not equal, the TLS connection is not established. If the CN uses a domain name, the certificate can also use wildcards ('*') to replace parts of the domain name. The valid range is a string of up to 49 characters. **Note:** This parameter is applicable only if the parameter PeerHostNameVerificationMode is set to 1 or 2.

6.4.5 SSH Parameters

The Secure Shell (SSH) parameters are described in the table below.

Table 6-22: SSH Parameters

Parameter	Description
[SSHAdminKey]	Determines the RSA public key for strong authentication to logging in to the Secure Shell (SSH) interface (if enabled). The value should be a base64-encoded string. The value can be a maximum length of 511 characters. For additional information, refer to the *Product Reference Manual*.

Parameter	Description
[SSHRequirePublicKey]	Enables or disables RSA public keys for SSH. ▪ **[0]** = RSA public keys are optional if a value is configured for the parameter SSHAdminKey (default). ▪ **[1]** = RSA public keys are mandatory.
Web/EMS: SSH Server Enable **[SSHServerEnable]**	Enables or disables the embedded Secure Shell (SSH) server. ▪ **[0]** Disable (default) ▪ **[1]** Enable
Web/EMS: SSH Server Port **[SSHServerPort]**	Defines the port number for the embedded SSH server. Range is any valid port number. The default port is 22.

6.4.6 IPSec Parameters

The Internet Protocol security (IPSec) parameters are described in the table below.

Table 6-23: IPSec Parameters

Parameter	Description
IPSec Parameters	
Web: Enable IP Security EMS: IPSec Enable **[EnableIPSec]**	Enables or disables IPSec on the device. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Note:** For this parameter to take effect, a device reset is required.
Web: IP Security Associations Table **EMS: IPSec SA Table**	
[IPSecSATable]	This *ini* file table parameter configures the IPSec SA table. This table allows you to configure the Internet Key Exchange (IKE) and IP Security (IPSec) protocols. You can define up to 20 IPSec peers. The format of this parameter is as follows: [IPsecSATable] FORMAT IPsecSATable_Index = IPsecSATable_RemoteEndpointAddressOrName, IPsecSATable_AuthenticationMethod, IPsecSATable_SharedKey, IPsecSATable_SourcePort, IPsecSATable_DestPort, IPsecSATable_Protocol, IPsecSATable_Phase1SaLifetimeInSec, IPsecSATable_Phase2SaLifetimeInSec, IPsecSATable_Phase2SaLifetimeInKB, IPsecSATable_DPDmode, IPsecSATable_IPsecMode, IPsecSATable_RemoteTunnelAddress, IPsecSATable_RemoteSubnetIPAddress, IPsecSATable_RemoteSubnetPrefixLength; [\IPsecSATable] For example: IPsecSATable 1 = 0, 10.3.2.73, 0, 123456789, 0, 0, 0, 0, 28800, 3600; In the above example, a single IPSec/IKE peer (10.3.2.73) is configured. Pre-shared key authentication is selected, with the pre-shared key set to 123456789. In addition, a lifetime of 28800 seconds is selected for IKE and a lifetime of 3600 seconds is selected for IPSec.

Parameter	Description
	Notes: ▪ Each row in the table refers to a different IP destination. ▪ To support more than one Encryption/Authentication proposal, for each proposal specify the relevant parameters in the Format line. ▪ The proposal list must be contiguous. ▪ For a detailed description of this table and to configure the table using the Web interface, refer to "Configuring the IP Security Associations Table" on page 96. ▪ For an explanation on using *ini* file table parameters, refer to "Configuring *ini* File Table Parameters" on page 222.
Web: IP Security Proposal Table **EMS: IPSec Proposal Table**	
[IPSecProposalTable]	This *ini* file table parameter configures up to four IKE proposal settings, where each proposal defines an encryption algorithm, an authentication algorithm, and a Diffie-Hellman group identifier. [IPsecProposalTable] FORMAT IPsecProposalTable_Index = IPsecProposalTable_EncryptionAlgorithm, IPsecProposalTable_AuthenticationAlgorithm, IPsecProposalTable_DHGroup; [\IPsecProposalTable] For example: IPsecProposalTable 0 = 3, 2, 1; IPsecProposalTable 1 = 2, 2, 1; In the example above, two proposals are defined: ▪ Proposal 0: AES, SHA1, DH group 2 ▪ Proposal 1: 3DES, SHA1, DH group 2 **Notes:** ▪ Each row in the table refers to a different IKE peer. ▪ To support more than one Encryption / Authentication / DH Group proposal, for each proposal specify the relevant parameters in the Format line. ▪ The proposal list must be contiguous. ▪ For a detailed description of this table and to configure the table using the Web interface, refer to "Configuring the IP Security Proposal Table" on page 95. ▪ For an explanation on using *ini* file table parameters, refer to "Configuring *ini* File Table Parameters" on page 222.

6.4.7 OCSP Parameters

The Online Certificate Status Protocol (OCSP) parameters are described in the table below.

Table 6-24: OCSP Parameters

Parameter	Description
EMS: OCSP Enable **[OCSPEnable]**	Enables or disables certificate checking using OCSP. ▪ **[0]** = Disable (default). ▪ **[1]** = Enable. For a description of OCSP, refer to the *Product Reference Manual*.
EMS: OCSP Server IP **[OCSPServerIP]**	Defines the IP address of the OCSP server. The default IP address is 0.0.0.0.
[OCSPSecondaryServerIP]	Defines the IP address (in dotted-decimal notation) of the secondary OCSP server (optional). The default IP address is 0.0.0.0.
EMS: OCSP Server Port **[OCSPServerPort]**	Defines the OCSP server's TCP port number. The default port number is 2560.
EMS: OCSP Default Response **[OCSPDefaultResponse]**	Determines the default OCSP behavior when the server cannot be contacted. ▪ **[0]** = Rejects peer certificate (default). ▪ **[1]** = Allows peer certificate.

6.5 RADIUS Parameters

The RADIUS parameters are described in the table below. For detailed information on the supported RADIUS attributes, refer to "Supported RADIUS Attributes" on page 596.

Table 6-25: RADIUS Parameters

Parameter	Description
Web: Enable RADIUS Access Control **[EnableRADIUS]**	Determines whether the RADIUS application is enabled. ▪ **[0]** Disable = RADIUS application is disabled (default). ▪ **[1]** Enable = RADIUS application is enabled. **Note:** For this parameter to take effect, a device reset is required.
Web: Accounting Server IP Address **[RADIUSAccServerIP]**	IP address of the RADIUS accounting server.
Web: Accounting Port **[RADIUSAccPort]**	Port of the RADIUS accounting server. The default value is 1646.
Web/EMS: RADIUS Accounting Type **[RADIUSAccountingType]**	Determines when the RADIUS accounting messages are sent to the RADIUS accounting server. ▪ **[0]** At Call Release = Sent at call release only (default). ▪ **[1]** At Connect & Release = Sent at call connect and release. ▪ **[2]** At Setup & Release = Sent at call setup and release.

Parameter	Description
Web: AAA Indications EMS: Indications **[AAAIndications]**	Determines the Authentication, Authorization and Accounting (AAA) indications. ▪ **[0]** None = No indications (default). ▪ **[3]** Accounting Only = Only accounting indications are used.
Web: Device Behavior Upon RADIUS Timeout **[BehaviorUponRadiusTimeout]**	Defines the device's response upon a RADIUS timeout. ▪ **[0]** Deny Access = Denies access. ▪ **[1]** Verify Access Locally = Checks password locally (default).
[MaxRADIUSSessions]	Number of concurrent calls that can communicate with the RADIUS server (optional). The valid range is 0 to 240. The default value is 240.
[RADIUSRetransmission]	Number of retransmission retries. The valid range is 1 to 10. The default value is 3.
[RadiusTO]	Determines the time interval (measured in seconds) the device waits for a response before a RADIUS retransmission is issued. The valid range is 1 to 30. The default value is 10.
Web: RADIUS Authentication Server IP Address **[RADIUSAuthServerIP]**	IP address of the RADIUS authentication server. **Note:** For this parameter to take effect, a device reset is required.
[RADIUSAuthPort]	RADIUS Authentication Server Port. **Note:** For this parameter to take effect, a device reset is required.
Web: RADIUS Shared Secret **[SharedSecret]**	'Secret' used to authenticate the device to the RADIUS server. This should be a cryptically strong password.
Web: Default Access Level **[DefaultAccessLevel]**	Defines the default access level for the device when the RADIUS (authentication) response doesn't include an access level attribute. The valid range is 0 to 255. The default value is 200 (Security Administrator').
Web: Local RADIUS Password Cache Mode **[RadiusLocalCacheMode]**	Defines the device's mode of operation regarding the timer (configured by the parameter RadiusLocalCacheTimeout) that determines the validity of the user name and password (verified by the RADIUS server). ▪ **[0]** Absolute Expiry Timer = when you access a Web page, the timeout doesn't reset, instead it continues decreasing. ▪ **[1]** Reset Timer Upon Access = upon each access to a Web page, the timeout always resets (reverts to the initial value configured by RadiusLocalCacheTimeout).
Web: Local RADIUS Password Cache Timeout **[RadiusLocalCacheTimeout]**	Defines the time (in seconds) the locally stored user name and password (verified by the RADIUS server) are valid. When this time expires, the user name and password become invalid and a must be re-verified with the RADIUS server. The valid range is 1 to 0xFFFFFF. The default value is 300 (5 minutes). ▪ **[-1]** = Never expires. ▪ **[0]** = Each request requires RADIUS authentication.

Parameter	Description
Web: RADIUS VSA Vendor ID **[RadiusVSAVendorID]**	Defines the vendor ID that the device accepts when parsing a RADIUS response packet. The valid range is 0 to 0xFFFFFFFF. The default value is 5003.
Web: RADIUS VSA Access Level Attribute **[RadiusVSAAccessAttribute]**	Defines the code that indicates the access level attribute in the Vendor Specific Attributes (VSA) section of the received RADIUS packet. The valid range is 0 to 255. The default value is 35.

6.6 SNMP Parameters

The SNMP parameters are described in the table below.

Table 6-26: SNMP Parameters

Parameter	Description
Web: Enable SNMP **[DisableSNMP]**	Determines whether SNMP is enabled. ▪ **[0]** Enable = SNMP is enabled (default). ▪ **[1]** Disable = SNMP is disabled and no traps are sent.
[SNMPPort]	The device's local UDP port used for SNMP Get/Set commands. The range is 100 to 3999. The default port is 161. **Note:** For this parameter to take effect, a device reset is required.
[SNMPTrustedMGR_x]	Defines up to five IP addresses of remote trusted SNMP managers from which the SNMP agent accepts and processes SNMP Get and Set requests. **Notes:** ▪ By default, the SNMP agent accepts SNMP Get and Set requests from any IP address, as long as the correct community string is used in the request. Security can be enhanced by using Trusted Managers, which is an IP address from which the SNMP agent accepts and processes SNMP requests. ▪ If no values are assigned to these parameters any manager can access the device. ▪ Trusted managers can work with all community strings.
[ChassisPhysicalAlias]	This object is an 'alias' name for the physical entity as specified by a network manager, and provides a non-volatile 'handle' for the physical entity. The valid range is a string of up to 255 characters.
[ChassisPhysicalAssetID]	This object is a user-assigned asset tracking identifier for the device's chassis as specified by an EMS, and provides non-volatile storage of this information. The valid range is a string of up to 255 characters.
[ifAlias]	The textual name of the interface. The value is equal to the ifAlias SNMP MIB object. The valid range is a string of up to 64 characters.

Parameter	Description
EMS: Keep Alive Trap Port **[KeepAliveTrapPort]**	The port to which the keep-alive traps are sent. The valid range is 0 - 65534. The default is port 162.
[SendKeepAliveTrap]	When enabled, this parameter invokes the keep-alive trap and sends it every 9/10 of the time defined in the parameter defining NAT Binding Default Timeout. ▪ **[0]** = Disable ▪ **[1]** = Enable **Note:** For this parameter to take effect, a device reset is required.
[SNMPSysOid]	Defines the base product system OID. The default is eSNMP_AC_PRODUCT_BASE_OID_D. **Note:** For this parameter to take effect, a device reset is required.
[SNMPTrapEnterpriseOid]	Defines a Trap Enterprise OID. The default is eSNMP_AC_ENTERPRISE_OID. The inner shift of the trap in the AcTrap subtree is added to the end of the OID in this parameter. **Note:** For this parameter to take effect, a device reset is required.
[acUserInputAlarmDescription]	Defines the description of the input alarm.
[acUserInputAlarmSeverity]	Defines the severity of the input alarm.
[AlarmHistoryTableMaxSize]	Determines the maximum number of rows in the Alarm History table. This parameter can be controlled by the Config Global Entry Limit MIB (located in the Notification Log MIB). The valid range is 50 to 1000. The default value is 500. **Note:** For this parameter to take effect, a device reset is required.
Web: SNMP Trap Destination Parameters **EMS: Network > SNMP Managers Table** **Note:** Up to five SNMP trap managers can be defined.	
SNMP Manager **[SNMPManagerIsUsed_x]**	Determines the validity of the parameters (IP address and port number) of the corresponding SNMP Manager used to receive SNMP traps. ▪ **[0]** (Check box cleared) = Disabled (default) ▪ **[1]** (Check box selected) = Enabled
Web: IP Address EMS: Address **[SNMPManagerTableIP_x]**	Defines the IP address of the remote host used as an SNMP Manager. The device sends SNMP traps to this IP address. Enter the IP address in dotted-decimal notation, e.g., 108.10.1.255.
Web: Trap Port EMS: Port **[SNMPManagerTrapPort_x]**	Defines the port number of the remote SNMP Manager. The device sends SNMP traps to this port. The valid SNMP trap port range is 100 to 4000. The default port is 162.
Web: Trap Enable **[SNMPManagerTrapSendingEnable_x]**	Activates or de-activates the sending of traps to the corresponding SNMP Manager. ▪ **[0]** Disable = Sending is disabled. ▪ **[1]** Enable = Sending is enabled (default).

Parameter	Description
[SNMPManagerTrapUser_x]	This parameter can be set to the name of any configured SNMPV3 user to associate with this trap destination. This determines the trap format, authentication level, and encryption level. By default, the trap is associated with the SNMP trap community string.
Web: Trap Manager Host Name **[SNMPTrapManagerHostName]**	Defines an FQDN of a remote host that is used as an SNMP manager. The resolved IP address replaces the last entry in the Trap Manager table (defined by the parameter SNMPManagerTableIP_x) and the last trap manager entry of snmpTargetAddrTable in the snmpTargetMIB. For example: 'mngr.corp.mycompany.com'. The valid range is a 99-character string.
SNMP Community String Parameters	
Community String **[SNMPReadOnlyCommunityString_x]**	Defines up to five read-only SNMP community strings (up to 19 characters each). The default string is 'public'.
Community String **[SNMPReadWriteCommunityString_x]**	Defines up to five read/write SNMP community strings (up to 19 characters each). The default string is 'private'.
Trap Community String **[SNMPTrapCommunityString]**	Community string used in traps (up to 19 characters). The default string is 'trapuser'.
Web: SNMP V3 Table **EMS: SNMP V3 Users**	
[SNMPUsers]	This *ini* file table parameter configures SNMP v3 users. The format of this parameter is as follows: [SNMPUsers] FORMAT SNMPUsers_Index = SNMPUsers_Username, SNMPUsers_AuthProtocol, SNMPUsers_PrivProtocol, SNMPUsers_AuthKey, SNMPUsers_PrivKey, SNMPUsers_Group; [\SNMPUsers] For example: SNMPUsers 1 = v3admin1, 1, 0, myauthkey, -, 1; The example above configures user 'v3admin1' with security level authNoPriv(2), authentication protocol MD5, authentication text password 'myauthkey', and ReadWriteGroup2. **Notes:** ▪ This parameter can include up to 10 indices. ▪ For a description of this table's individual parameters and for configuring the table using the Web interface, refer to "Configuring SNMP V3 Users" on page 193. ▪ For an explanation on using *ini* file table parameters, refer to "Configuring *ini* File Table Parameters" on page 222

6.7 SIP Configuration Parameters

This subsection describes the device's SIP parameters.

6.7.1 General SIP Parameters

The general SIP parameters are described in the table below.

Table 6-27: General SIP Parameters

Parameter	Description
[SIPForceRport]	Determines whether the device sends SIP responses to the UDP port from where SIP requests are received even if the 'rport' parameter is not present in the SIP Via header. ▪ **[0]** (default) = Disabled - the device sends the SIP response to the UDP port defined in the Via header. If the Via header contains the 'rport' parameter, the response is sent to the UDP port from where the SIP request is received. ▪ **[1]** = Enabled - SIP responses are sent to the UDP port from where SIP requests are received even if the 'rport' parameter is not present in the Via header.
Web: Max Number of Active Calls EMS: Maximum Concurrent Calls **[MaxActiveCalls]**	Defines the maximum number of simultaneous active calls supported by the device. If the maximum number of calls is reached, new calls are not established. The valid range is 1 to the maximum number of supported channels. The default value is the maximum available channels (i.e., no restriction on the maximum number of calls).
Web/EMS: PRACK Mode **[PrackMode]**	PRACK (Provisional Acknowledgment) mechanism mode for SIP 1xx reliable responses. ▪ **[0]** Disable ▪ **[1]** Supported (default) ▪ **[2]** Required **Notes:** ▪ The Supported and Required headers contain the '100rel' tag. ▪ The device sends PRACK messages if 180/183 responses are received with '100rel' in the Supported or Required headers.
Web/EMS: Enable Early Media **[EnableEarlyMedia]**	Enables the device to send a 183 Session Progress response with SDP instead of a 180 Ringing, allowing the media stream to be established prior to the answering of the call. ▪ **[0]** Disable = Early Media is disabled (default). ▪ **[1]** Enable = Enables Early Media. For Analog interfaces: Note that to send a 183 response, you must also set the parameter ProgressIndicator2IP to 1. If it is equal to 0, 180 Ringing response is sent. For Digital interfaces: Sending a 183 response depends on the ISDN Progress Indicator (PI). It is sent only if PI is set to 1 or 8 in the received Proceeding or Alerting PRI messages. For CAS protocol, see the ProgressIndicator2IP parameter.

Parameter	Description
	Note: You can also configure early SIP 183 response immediately upon receipt of an INVITE, using the EnableEarly183 parameter.
Web/EMS: Enable Early 183 **[EnableEarly183]**	Determines whether the device sends a SIP 183 response with SDP to the IP immediately upon receipt of an INVITE message (for IP-to-Tel calls). The device sends the RTP packets only once it receives an ISDN Progress, Alerting with Progress indicator, or Connect message from the PSTN. ▪ **[0]** Disable (default) ▪ **[1]** Enable For example, if enabled and the device receives an ISDN Progress message, it starts sending RTP packets according to the initial negotiation without sending the 183 response again. Therefore, this feature reduces clipping of early media. **Notes:** ▪ To enable this feature, configure the EnableEarlyMedia parameter to 1. ▪ This feature is applicable only to ISDN interfaces.
Web: 183 Message Behavior EMS: SIP 183 Behaviour **[SIP183Behaviour]**	Digital interfaces: Defines the ISDN message that is sent when the 183 Session Progress message is received for IP-to-Tel calls. Analog interfaces: Defines the response of the device upon receipt of a SIP 183 response. ▪ **[0]** Progress = Digital interfaces: The device sends a Progress message. Analog interfaces: A 183 response (without SDP) does not cause the device to play a ringback tone (default). ▪ **[1]** Alert = Digital interfaces: The device sends an Alerting message (upon receipt of a 183 response) instead of an ISDN Progress message. Analog interfaces: 183 response is handled by the device as if a 180 Ringing response is received, and the device plays a ringback tone.
Web: Session-Expires Time EMS: Sip Session Expires **[SIPSessionExpires]**	Determines the numerical value that is sent in the Session-Expires header in the first INVITE request or response (if the call is answered). The valid range is 1 to 86,400 sec. The default is 0 (i.e., the Session-Expires header is disabled).
Web: Minimum Session-Expires EMS: Minimal Session Refresh Value **[MinSE]**	Defines the time (in seconds) that is used in the Min-SE header. This header defines the minimum time that the user agent refreshes the session. The valid range is 10 to 100,000. The default value is 90.

Parameter	Description
Web/EMS: Session Expires Method **[SessionExpiresMethod]**	Determines the SIP method used for session-timer updates. ▪ **[0]** Re-INVITE = Uses Re-INVITE messages for session-timer updates (default). ▪ **[1]** UPDATE = Uses UPDATE messages. **Notes:** ▪ The device can receive session-timer refreshes using both methods. ▪ The UPDATE message used for session-timer is excluded from the SDP body.
[RemoveToTagInFailureResponse]	Determines whether the device removes the 'to' header tag from final SIP failure responses to INVITE transactions. ▪ **[0]** = Do not remove tag (default). ▪ **[1]** = Remove tag.
[EnableRTCPAttribute]	Enables or disables the use of the 'rtcp' attribute in the outgoing SDP. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable
EMS: Options User Part **[OPTIONSUserPart]**	Defines the user part value of the Request-URI for outgoing SIP OPTIONS requests. If no value is configured, the endpoint number (analog interfaces) or configuration parameter 'Username' value (digital interfaces) is used. A special value is 'empty', indicating that no user part in the Request-URI (host part only) is used. The valid range is a 30-character string. The default value is an empty string ('').
Web: TDM Over IP Minimum Calls For Trunk Activation EMS: TDM Over IP Min Calls For Trunk Activation **[TDMOverIPMinCallsForTrunkActivation]**	Defines the minimal number of SIP dialogs that must be established when using TDM Tunneling to consider the specific trunk as active. When using TDM Tunneling, if calls from this defined number of B-channels pertaining to a specific Trunk fail (i.e., SIP dialogs are not correctly set up), an AIS alarm is sent on this trunk toward the PSTN and all current calls are dropped. The originator gateway continues the INVITE attempts. When this number of calls succeed (i.e., SIP dialogs are correctly set up), the AIS alarm is cleared. The valid range is 0 to 31. The default value is 0 (i.e., don't send AIS alarms).
[TDMoIPInitiateInviteTime]	Determines the time (in msec) between the first INVITE issued within the same trunk when implementing the TDM tunneling application. The valid value range is 500 to 1000. The default is 500.
[TDMoIPInviteRetryTime]	Determines the time (in msec) between call release and a new INVITE when implementing the TDM tunneling application. The valid value range is 10,000 to 20,000. The default is 10,000.

Parameter	Description
Web: Fax Signaling Method EMS: Fax Used **[IsFaxUsed]**	Determines the SIP signaling method for establishing and transmitting a fax session after a fax is detected. ▪ **[0]** No Fax = No fax negotiation using SIP signaling. Fax transport method is according to the parameter FaxTransportMode (default). ▪ **[1]** T.38 Relay = Initiates T.38 fax relay. ▪ **[2]** G.711 Transport = Initiates fax/modem using the coder G.711 A-law/Mu-law with adaptations (refer to Note below). ▪ **[3]** Fax Fallback = Initiates T.38 fax relay. If the T.38 negotiation fails, the device re-initiates a fax session using the coder G.711 A-law/μ-law with adaptations (refer to the Note below). **Notes:** ▪ Fax adaptations (for options 2 and 3): ✓ Echo Canceller = On ✓ Silence Compression = Off ✓ Echo Canceller Non-Linear Processor Mode = Off ✓ Dynamic Jitter Buffer Minimum Delay = 40 ✓ Dynamic Jitter Buffer Optimization Factor = 13 ▪ If the device initiates a fax session using G.711 (option 2 and possibly 3), a 'gpmd' attribute is added to the SDP in the following format: ✓ **For A-law:** 'a=gpmd:8 vbd=yes;ecan=on' ✓ **For μ-law:** 'a=gpmd:0 vbd=yes;ecan=on' ▪ When this parameter is set to 1, 2, or 3, the parameter FaxTransportMode is ignored. ▪ When this parameter is set to 0, T.38 might still be used without the control protocol's involvement. To completely disable T.38, set FaxTransportMode to a value other than 1. ▪ For detailed information on fax transport methods, refer to "Fax/Modem Transport Modes" on page 556.
Web: SIP Transport Type EMS: Transport Type **[SIPTransportType]**	Determines the default transport layer for outgoing SIP calls initiated by the device. ▪ **[0]** UDP (default) ▪ **[1]** TCP ▪ **[2]** TLS (SIPS) **Notes:** ▪ It's recommended to use TLS for communication with a SIP Proxy and not for direct device-to-device communication. ▪ For received calls (i.e., incoming), the device accepts all these protocols. ▪ The value of this parameter is also used by the SAS application as the default transport layer for outgoing SIP calls.
Web: SIP UDP Local Port EMS: Local SIP Port **[LocalSIPPort]**	Local UDP port for SIP messages. The valid range is 1 to 65534. The default value is 5060.

Parameter	Description
Web: SIP TCP Local Port EMS: TCP Local SIP Port **[TCPLocalSIPPort]**	Local TCP port for SIP messages. The valid range is 1 to 65535. The default value is 5060.
Web: SIP TLS Local Port EMS: TLS Local SIP Port **[TLSLocalSIPPort]**	Local TLS port for SIP messages. The valid range is 1 to 65535. The default value is 5061. **Note:** The value of this parameter must be different from the value of the parameter TCPLocalSIPPort.
Web/EMS: Enable SIPS **[EnableSIPS]**	Enables secured SIP (SIPS URI) connections over multiple hops. ▪ **[0]** Disable (default). ▪ **[1]** Enable. When the parameter SIPTransportType is set to 2 (i.e., TLS) and the parameter EnableSIPS is disabled, TLS is used for the next network hop only. When the parameter SIPTransportType is set to 2 or 1 (i.e., TCP or TLS) and EnableSIPS is enabled, TLS is used through the entire connection (over multiple hops). **Note:** If this parameter is enabled and the parameter SIPTransportType is set to 0 (i.e., UDP), the connection fails.
Web/EMS: Enable TCP Connection Reuse **[EnableTCPConnectionReuse]**	Enables the reuse of the same TCP connection for all calls to the same destination. ▪ **[0]** Disable = Use a separate TCP connection for each call. ▪ **[1]** Enable = Use the same TCP connection for all calls (default).
Web/EMS: Reliable Connection Persistent Mode **[ReliableConnectionPersistentMode]**	Determines whether all TCP/TLS connections are set as persistent and therefore, not released. ▪ **[0]** = Disable (default) - all TCP connections (except those that are set to a proxy IP) are released if not used by any SIP dialog\transaction. ▪ **[1]** = Enable - TCP connections to all destinations are persistent and not released unless the device reaches 70% of its maximum TCP resources. While trying to send a SIP message connection, reuse policy determines whether alive connections to the specific destination are re-used. Persistent TCP connection ensures less network traffic due to fewer setting up and tearing down of TCP connections and reduced latency on subsequent requests due to avoidance of initial TCP handshake. For TLS, persistent connection may reduce the number of costly TLS handshakes to establish security associations, in addition to the initial TCP connection set up. **Note:** If the destination is a Proxy server, the TCP/TLS connection is persistent regardless of the settings of this parameter.
Web/EMS: TCP Timeout **[SIPTCPTimeout]**	Defines the Timer B (INVITE transaction timeout timer) and Timer F (non-INVITE transaction timeout timer), as defined in RFC 3261, when the SIP Transport Type is TCP. The valid range is 0 to 40 sec. The default value is 64*SIPT1Rtx msec.

Parameter	Description
Web: SIP Destination Port EMS: Destination Port **[SIPDestinationPort]**	SIP destination port for sending initial SIP requests. The valid range is 1 to 65534. The default port is 5060. **Note:** SIP responses are sent to the port specified in the Via header.
Web: Use user=phone in SIP URL EMS: Is User Phone **[IsUserPhone]**	Determines whether the 'user=phone' string is added to the SIP URI and SIP To header. ▪ **[0]** No = 'user=phone' string is not added. ▪ **[1]** Yes = 'user=phone' string is part of the SIP URI and SIP To header (default).
Web: Use user=phone in From Header EMS: Is User Phone In From **[IsUserPhoneInFrom]**	Determines whether the 'user=phone' string is added to the From and Contact SIP headers. ▪ **[0]** No = Doesn't add 'user=phone' string (default). ▪ **[1]** Yes = 'user=phone' string is part of the From and Contact headers.
Web: Use Tel URI for Asserted Identity **[UseTelURIForAssertedID]**	Determines the format of the URI in the P-Asserted-Identity and P-Preferred-Identity headers. ▪ **[0]** Disable = 'sip:' (default) ▪ **[1]** Enable = 'tel:'
Web: Tel to IP No Answer Timeout EMS: IP Alert Timeout **[IPAlertTimeout]**	Defines the time (in seconds) that the device waits for a 200 OK response from the called party (IP side) after sending an INVITE message. If the timer expires, the call is released. The valid range is 0 to 3600. The default value is 180.
Web: Enable Remote Party ID EMS: Enable RPI Header **[EnableRPIheader]**	Enables Remote-Party-Identity headers for calling and called numbers for Tel-to-IP calls. ▪ **[0]** Disable (default). ▪ **[1]** Enable = Remote-Party-Identity headers are generated in SIP INVITE messages for both called and calling numbers.

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>Web: Enable History-Info Header
EMS: Enable History Info
[EnableHistoryInfo]</td><td>
Enables usage of the History-Info header.

[0] Disable (default)
[1] Enable

User Agent Client (UAC) Behavior:

Initial request: The History-Info header is equal to the Request-URI. If a PSTN Redirect number is received, it is added as an additional History-Info header with an appropriate reason.
Upon receiving the final failure response, the device copies the History-Info as is, adds the reason of the failure response to the last entry, and concatenates a new destination to it (if an additional request is sent). The order of the reasons is as follows:
<ol type="a">
Q.850 Reason
SIP Reason
SIP Response code

Upon receiving the final response (success or failure), the device searches for a Redirect reason in the History-Info (i.e., 3xx/4xx SIP reason). If found, it is passed to ISDN according to the following table:

<table>
<tr><th>SIP Reason Code</th><th>ISDN Redirecting Reason</th></tr>
<tr><td>302 - Moved Temporarily</td><td>Call Forward Universal (CFU)</td></tr>
<tr><td>408 - Request Timeout</td><td rowspan="3">Call Forward No Answer (CFNA)</td></tr>
<tr><td>480 - Temporarily Unavailable</td></tr>
<tr><td>487 - Request Terminated</td></tr>
<tr><td>486 - Busy Here</td><td rowspan="2">Call Forward Busy (CFB)</td></tr>
<tr><td>600 - Busy Everywhere</td></tr>
</table>

If history reason is a Q.850 reason, it is translated to the SIP reason (according to the SIP-ISDN tables) and then to ISDN Redirect reason according to the table above.

User Agent Server (UAS) Behavior:

The History-Info header is sent only in the final response.
Upon receiving a request with History-Info, the UAS checks the policy in the request. If a 'session', 'header', or 'history' policy tag is found, the (final) response is sent without History-Info; otherwise, it is copied from the request.

</td></tr>
<tr><td>Web: Use Tgrp Information
EMS: Use SIP Tgrp
[UseSIPTgrp]</td><td>
Determines whether the SIP 'tgrp' parameter is used. This SIP parameter specifies the Trunk Group to which the call belongs (according to RFC 4904). For example, the SIP message below indicates that the call belongs to Trunk Group ID 1:
INVITE sip::+16305550100;tgrp=1;trunk-context=example.com@10.1.0.3;user=phone SIP/2.0

[0] Disable (default) = The 'tgrp' parameter isn't used.
[1] Send Only = The Trunk Group number is added to the 'tgrp' parameter value in the Contact header of outgoing SIP messages. If a Trunk Group number is not associated with the call, the 'tgrp' parameter isn't included. If a 'tgrp' value is specified in incoming messages, it is ignored.

</td></tr>
</table>

Parameter	Description
	▪ **[2]** Send and Receive = The functionality of outgoing SIP messages is identical to the functionality described in option 1. In addition, for incoming SIP INVITEs, if the Request-URI includes a 'tgrp' parameter, the device routes the call according to that value (if possible). The Contact header in the outgoing SIP INVITE (Tel-to-IP call) contains "tgrp=<source trunk group ID>;trunk-context=<gateway IP address>". The <source trunk group ID> is the Trunk Group ID where incoming calls from Tel is received. For IP-Tel calls, the SIP 200 OK device's response contains "tgrp=<destination trunk group ID>;trunk-context=<gateway IP address>". The <destination trunk group ID> is the Trunk Group ID used for outgoing Tel calls. The <gateway IP address> in "trunk-context" can be configured using the parameter SIPGatewayName. ▪ **[3]** UCR 2008 = Interworks the hotline "Off Hook Indicator" parameter between SIP and ISDN: ✓ For IP-to-ISDN calls: - The device interworks the SIP tgrp=hotline parameter (received in INVITE) to ISDN Setup with the Off Hook Indicator IE of "Voice", and "Speech" Bearer Capability IE. Note that the Off Hook Indicator IE is described in UCR 2008 specifications. - The device interworks the SIP tgrp=hotline-ccdata parameter (received in INVITE) to ISDN Setup with an Off Hook Indicator IE of "Data", and with "Unrestricted 64k" Bearer Capability IE. The following is an example of the INVITE with tgrp=hotline-ccdata: INVITE sip:1234567;tgrp=hotline-ccdata;trunk-context=dsn.mil@example.com ✓ For ISDN-to-IP calls: - The device interworks ISDN Setup with an Off Hook Indicator of "Voice" to SIP INVITE with "tgrp=hotline;trunk-context=dsn.mil" in the Contact header. - The device interworks ISDN Setup with an Off Hook indicator of "Data" to SIP INVITE with "tgrp=hotline-ccdata;trunk-context=dsn.mil" in the Contact header. - If ISDN Setup does not contain an Off Hook Indicator IE and the Bearer Capability IE contains "Unrestricted 64k", the outgoing INVITE includes "tgrp=ccdata;trunk-context=dsn.mil". If the Bearer Capability IE contains "Speech", the INVITE in this case does not contain tgrp and trunk-context parameters. **Note:** IP-to-Tel configuration (using the parameter PSTNPrefix) overrides the 'tgrp' parameter in incoming INVITE messages.
Web/EMS: TGRP Routing Precedence **[TGRProutingPrecedence]**	Determines the precedence method for routing IP-to-Tel calls - according to the 'Inbound IP Routing Table' or according to the SIP 'tgrp' parameter. ▪ **[0]** (default) = IP-to-Tel routing is determined by the 'Inbound IP Routing Table' (PSTNPrefix parameter). If a matching rule is not found in this table, the device uses the Trunk Group parameters for routing the call. ▪ **[1]** = The device first places precedence on the 'tgrp'

Parameter	Description
	parameter for IP-to-Tel routing. If the received INVITE Request-URI does not contain the 'tgrp' parameter or if the Trunk Group number is not defined, then the 'Inbound IP Routing Table' is used for routing the call. Below is an example of an INVITE Request-URI with the 'tgrp' parameter, indicating that the IP call should be routed to Trunk Group 7: INVITE sip:200;tgrp=7;trunk-context=example.com@10.33.2.68;user=phone SIP/2.0 **Notes:** ▪ For enabling routing based on the 'tgrp' parameter, the UseSIPTgrp parameter must be set to 2. ▪ For IP-to-Tel routing based on the 'dtg' parameter (instead of the 'tgrp' parameter), use the parameter UseBroadsoftDTG.
[UseBroadsoftDTG]	Determines whether the device uses the 'dtg' parameter for routing IP-to-Tel calls to a specific Trunk Group. ▪ **[0]** Disable (default) ▪ **[1]** Enable When this parameter is enabled, if the Request-URI in the received SIP INVITE includes the 'dtg' parameter, the device routes the call to the Trunk Group according to its value. This parameter is used instead of the 'tgrp/trunk-context' parameters. The 'dtg' parameter appears in the INVITE Request-URI (and in the To header). For example, the received SIP message below routes the call to Trunk Group ID 56: INVITE sip:123456@192.168.1.2;dtg=56;user=phone SIP/2.0 **Note:** If the Trunk Group is not found based on the 'dtg' parameter, the 'Inbound IP Routing Table' is used instead for routing the call to the appropriate Trunk Group.
Web/EMS: Enable GRUU **[EnableGRUU]**	Determines whether the Globally Routable User Agent URIs (GRUU) mechanism is used. ▪ **[0]** Disable (default) ▪ **[1]** Enable The device obtains a GRUU by generating a normal REGISTER request. This request contains a Supported header with the value 'gruu'. The device includes a '+sip.instance' Contact header parameter for each contact for which the GRUU is desired. This Contact parameter contains a globally unique ID that identifies the device instance. The global unique ID is as follows: ▪ If registration is per endpoint (i.e., the parameter AuthenticationMode is set to 0) it is the MAC address of the device concatenated with the phone number of the endpoint. ▪ If the registration is per device (i.e., the parameter AuthenticationMode is set to 1) it is only the MAC address. ▪ When the User Information mechanism is used, the globally unique ID is the MAC address concatenated with the phone number of the endpoint (defined in the User Info file).

Parameter	Description

	If the Registrar/Proxy supports GRUU, the REGISTER responses contain the 'gruu' parameter in each Contact header field. The Registrar/Proxy provides the same GRUU for the same AOR and instance-id in case of sending REGISTER again after expiration of the registration.
	The device places the GRUU in any header field which contains a URI. It uses the GRUU in the following messages: INVITE requests, 2xx responses to INVITE, SUBSCRIBE requests, 2xx responses to SUBSCRIBE, NOTIFY requests, REFER requests, and 2xx responses to REFER.
	Note: If the GRUU contains the 'opaque' URI parameter, the device obtains the AOR for the user by stripping the parameter. The resulting URI is the AOR, for example:
	`AOR: sip:alice@example.com`
	`GRUU:`
	`sip:alice@example.com;opaque="kjh29x97us97d"`

EMS: Is CISCO Sce Mode **[IsCiscoSCEMode]**	Determines whether a Cisco gateway exists at the remote side. ▪ **[0]** = No Cisco gateway exists at the remote side (default). ▪ **[1]** = A Cisco gateway exists at the remote side. When a Cisco gateway exists at the remote side, the device must set the value of the 'annexb' parameter of the fmtp attribute in the SDP to 'no'. This logic is used if the parameter EnableSilenceCompression is set to 2 (enable without adaptation). In this case, Silence Suppression is used on the channel but not declared in the SDP. **Note:** The IsCiscoSCEMode parameter is applicable only when the selected coder is G.729.

Web: User-Agent Information
EMS: User Agent Display Info
[UserAgentDisplayInfo]
	Defines the string that is used in the SIP User-Agent and Server response headers. When configured, the string '<value for UserAgentDisplayInfo>/software version' is used, for example:
	`User-Agent: myproduct/v.6.00.010.006`
	If not configured, the default string, 'AudioCodes product-name s/w-version' is used, for example:
	`User-Agent: Audiocodes-Sip-Gateway-Mediant 1000/v.6.00.010.006`
	The maximum string length is 50 characters.
	Note: The software version number can't be modified.

Web/EMS: SDP Session Owner **[SIPSDPSessionOwner]**	Determines the value of the Owner line ('o' field) in outgoing SDP messages. The valid range is a string of up to 39 characters. The default value is 'AudiocodesGW'. For example: `o=AudiocodesGW 1145023829 1145023705 IN IP4 10.33.4.126`
Web/EMS: Subject **[SIPSubject]**	Defines the value of the Subject header in outgoing INVITE messages. If not specified, the Subject header isn't included (default). The maximum length is up to 50 characters.

Parameter	Description
Web: Multiple Packetization Time Format EMS: Multi Ptime Format **[MultiPtimeFormat]**	Determines whether the 'mptime' attribute is included in the outgoing SDP. ▪ **[0]** None = Disabled (default) ▪ **[1]** PacketCable = includes the 'mptime' attribute in the outgoing SDP - PacketCable-defined format The 'mptime' attribute enables the device to define a separate Packetization period for each negotiated coder in the SDP. The 'mptime' attribute is only included if this parameter is enabled, even if the remote side includes it in the SDP offer. Upon receipt, each coder receives its 'ptime' value in the following precedence: from 'mptime' attribute, from 'ptime' attribute, and then from default value.
EMS: Enable P Time **[EnablePtime]**	Determines whether the 'ptime' attribute is included in the SDP. ▪ **[0]** = Remove the 'ptime' attribute from SDP. ▪ **[1]** = Include the 'ptime' attribute in SDP (default).
Web/EMS: 3xx Behavior **[3xxBehavior]**	Determines the device's behavior regarding call identifiers when a 3xx response is received for an outgoing INVITE request. The device can either use the same call identifiers (Call-ID, Branch, To, and From tags) or change them in the new initiated INVITE. ▪ **[0]** Forward = Use different call identifiers for a redirected INVITE message (default). ▪ **[1]** Redirect = Use the same call identifiers.
Web/EMS: Enable P-Charging Vector **[EnablePChargingVector]**	Enables the inclusion of the P-Charging-Vector header to all outgoing INVITE messages. ▪ **[0]** Disable (default) ▪ **[1]** Enable
Web/EMS: Retry-After Time **[RetryAfterTime]**	Determines the time (in seconds) used in the Retry-After header when a 503 (Service Unavailable) response is generated by the device. The time range is 0 to 3,600. The default value is 0.
Web/EMS: Fake Retry After **[sec]** **[FakeRetryAfter]**	Determines whether the device, upon receipt of a SIP 503 response without a Retry-After header, behaves as if the 503 response included a Retry-After header and with the period (in seconds) specified by this parameter. ▪ **[0]** Disable ▪ Any positive value (in seconds) for defining the period When enabled, this feature allows the device to operate with Proxy servers that do not include the Retry-After SIP header in SIP 503 (Service Unavailable) responses to indicate an unavailable service. The Retry-After header is used with the 503 (Service Unavailable) response to indicate how long the service is expected to be unavailable to the requesting SIP client. The device maintains a list of available proxies, by using the Keep-Alive mechanism. The device checks the availability of proxies by sending SIP OPTIONS every keep-alive timeout to all proxies. If the device receives a SIP 503 response to an INVITE, it also marks that the proxy is out of service for the defined "Retry-After" period.

Parameter	Description
Web/EMS: Enable P-Associated-URI Header **[EnablePAssociatedURIHeader]**	Determines the device usage of the P-Associated-URI header. This header can be received in 200 OK responses to REGISTER requests. When enabled, the first URI in the P-Associated-URI header is used in subsequent requests as the From/P-Asserted-Identity headers value. ▪ **[0]** Disable (default). ▪ **[1]** Enable. **Note:** P-Associated-URIs in registration responses is handled only if the device is registered per endpoint (using the User Information file).
Web/EMS: Source Number Preference **[SourceNumberPreference]**	Determines the SIP header used for the source number in incoming INVITE messages. ▪ **''** = (empty string) Use the device's internal logic for header preference (default). The logic for filling the calling party parameters is as follows: the SIP header is selected first from which the calling party parameters are obtained: first priority is P-Asserted-Identity, second is Remote-Party-ID, and third is the From header. Once a URL is selected, all the calling party parameters are set from this header. If P-Asserted-Identity is selected, the Privacy header is checked and if the Privacy is set to 'id', the calling number is assumed restricted. ▪ **'FROM'** = Use the source number received in the From header.
[SelectSourceHeaderForCalledNumber]	Determines the SIP header used for obtaining the called number (destination) for IP-to-Tel calls. ▪ **[0]** Request-URI header (default) = Obtains the destination number from the user part of the Request-URI. ▪ **[1]** To header = Obtains the destination number from the user part of the To header. ▪ **[2]** P-Called-Party-ID header = Obtains the destination number from the P-Called-Party-ID header.
Web/EMS: Forking Handling Mode **[ForkingHandlingMode]**	Determines how the device handles the receipt of multiple SIP 18x responses when forking is used by a Proxy, for Tel-to-IP calls. ▪ **[0]** Parallel handling = The device opens a voice stream toward the first 18x SIP response that includes an SDP and disregards any 18x response with an SDP received thereafter (default). ▪ **[1]** Sequential handling = The device opens a voice stream toward the first 18x SIP response that includes an SDP and re-opens the stream toward any subsequent 18x responses with an SDP. **Note:** Regardless of this parameter value, once a SIP 200 OK response is received, the device uses the RTP information and re-opens the voice stream, if necessary.
Web: Forking Timeout **[ForkingTimeOut]**	The timeout (in seconds) that is started after the first SIP 2xx response has been received for a User Agent when a Proxy server performs call forking (Proxy server forwards the INVITE to multiple SIP User Agents). The device sends a SIP ACK and

Parameter	Description
	BYE in response to any additional SIP 2xx received from the Proxy within this timeout. Once this timeout elapses, the device ignores any subsequent SIP 2xx. The number of supported forking calls per channel is 20. In other words, for an INVITE message, the device can receive up to 20 forking responses from the Proxy server. The valid range is 0 to 30. The default is 30.
Web/EMS: Enable Reason Header **[EnableReasonHeader]**	Enables or disables the usage of the SIP Reason header. ▪ **[0]** Disable ▪ **[1]** Enable (default)
Web/EMS: Gateway Name **[SIPGatewayName]**	Assigns a name to the device (e.g., 'device123.com'). Ensure that the name you choose is the one with which the Proxy is configured to identify the device. **Note:** If specified, the device name is used as the host part of the SIP URI in the From header. If not specified, the device's IP address is used instead (default).
[ZeroSDPHandling]	Determines the device's response to an incoming SDP that includes an IP address of 0.0.0.0 in the SDP's Connection Information field (i.e., "c=IN IP4 0.0.0.0"). ▪ **[0]** = Sets the IP address of the outgoing SDP's c= field to 0.0.0.0 (default). ▪ **[1]** = Sets the IP address of the outgoing SDP c= field to the IP address of the device. If the incoming SDP doesn't contain the "a=inactive" line, the returned SDP contains the "a=recvonly" line.
Web/EMS: Enable Delayed Offer **[EnableDelayedOffer]**	Determines whether the device sends the initial INVITE message with or without an SDP. Sending the first INVITE without SDP is typically done by clients for obtaining the far-end's full list of capabilities before sending their own offer. (An alternative method for obtaining the list of supported capabilities is by using SIP OPTIONS, which is not supported by every SIP agent.) ▪ **[0]** Disable = The device sends the initial INVITE message with an SDP (default). ▪ **[1]** Enable = The device sends the initial INVITE message without an SDP.
Web/EMS: Enable Contact Restriction **[EnableContactRestriction]**	Determines whether the device sets the Contact header of outgoing INVITE requests to 'anonymous' for restricted calls. ▪ **[0]** Disable (default) ▪ **[1]** Enable
[AnonymousMode]	Determines whether the device's IP address is used as the URI host part instead of "anonymous.invalid" in the INVITE's From header for Tel-to-IP calls. ▪ **[0]** = (default) If the device receives a call from the Tel with blocked caller ID, it sends an INVITE with From: "anonymous"<anonymous@anonymous.invalid> ▪ **[1-]** = The device's IP address is used as the URI host part instead of "anonymous.invalid".

Parameter	Description
	This parameter may be useful, for example, for service providers who identify their SIP Trunking customers by their source phone number or IP address, reflected in the From header of the SIP INVITE. Therefore, even customers blocking their Caller ID can be identified by the service provider. Typically, if the device receives a call with blocked Caller ID from the PSTN side (e.g., Trunk connected to a PBX), it sends an INVITE to the IP with a From header as follows: From: “anonymous” <anonymous@anonymous.invalid>. This is in accordance with RFC 3325. However, when this parameter is set to 1, the device replaces the "anonymous.invalid" with its IP address.
EMS: P Asserted User Name **[PAssertedUserName]**	Defines a 'representative number' (up to 50 characters) that is used as the user part of the Request-URI in the P-Asserted-Identity header of an outgoing INVITE (for Tel-to-IP calls). The default value is null.
EMS: Use URL In Refer To Header **[UseAORInReferToHeader]**	Defines the source for the SIP URI set in the Refer-To header of outgoing REFER messages. ▪ **[0]** = Use SIP URI from Contact header of the initial call (default). ▪ **[1]** = Use SIP URI from To/From header of the initial call.
Web: Enable User-Information Usage **[EnableUserInfoUsage]**	Enables or disables the usage of the User Information, which is loaded to the device in the User Information auxiliary file. (For a description on User Information, refer to "Loading Auxiliary Files" on page 200.) ▪ **[0]** Disable (default). ▪ **[1]** Enable
[HandleReasonHeader]	Determines whether the device uses the value of the incoming SIP Reason header for Release Reason mapping. ▪ **[0]** Disregard Reason header in incoming SIP messages. ▪ **[1]** Use the Reason header value for Release Reason mapping (default).
[EnableSilenceSuppInSDP]	Determines the device's behavior upon receipt of SIP Re-INVITE messages that include the SDP's 'silencesupp:off' attribute. ▪ **[0]** = Disregard the 'silecesupp' attribute (default). ▪ **[1]** = Handle incoming Re-INVITE messages that include the 'silencesupp:off' attribute in the SDP as a request to switch to the Voice-Band-Data (VBD) mode. In addition, the device includes the attribute 'a=silencesupp:off' in its SDP offer. **Note:** This parameter is applicable only if the G.711 coder is used.
[EnableRport]	Enables or disables the usage of the 'rport' parameter in the Via header. ▪ **[0]** = Enabled. ▪ **[1]** = Disabled (default). The device adds an 'rport' parameter to the Via header of each outgoing SIP message. The first Proxy that receives this

Parameter	Description
	message sets the 'rport' value of the response to the actual port from where the request was received. This method is used, for example, to enable the device to identify its port mapping outside a NAT. If the Via header doesn't include the 'rport' parameter, the destination port of the response is obtained from the host part of the Via header. If the Via header includes the 'rport' parameter without a port value, the destination port of the response is the source port of the incoming request. If the Via header includes 'rport' with a port value (e.g., rport=1001), the destination port of the response is the port indicated in the 'rport' parmeter.
Web: Enable X-Channel Header EMS: X Channel Header **[XChannelHeader]**	Determines whether the SIP X-Channel header is added to SIP messages for providing information on the physical Trunk/B-channel on which the call is received or placed. ▪ **[0]** Disable = X-Channel header is not used (default). ▪ **[1]** Enable = X-Channel header is generated by the device and sent in INVITE messages and 180, 183, and 200 OK SIP responses. The header includes the Trunk number, B-channel, and the device's IP address. For example, 'x-channel: DS/DS1-5/8;IP=192.168.13.1', where: ✓ 'DS/DS-1' is a constant string ✓ '5' is the Trunk number ✓ '8' is the B-channel ✓ 'IP=192.168.13.1' is the device's IP address
Web/EMS: Progress Indicator to IP **[ProgressIndicator2IP]**	For Analog (FXS/FXO) interfaces: ▪ **[-1**] Not Configured (default) = Default values are used. The default for FXO interfaces is 1; The default for FXS interfaces is 0. ▪ **[0]** No PI = For IP-to-Tel calls, the device sends a 180 Ringing response to IP after placing a call to a phone (FXS) or PBX (FXO). ▪ **[1]** PI = 1, **[8]** PI = 8: For IP-to-Tel calls, if the parameter EnableEarlyMedia is set to 1, the device sends a 183 Session Progress message with SDP immediately after a call is placed to a phone/PBX. This is used to cut-through the voice path before the remote party answers the call. This allows the originating party to listen to network Call Progress Tones (such as ringback tone or other network announcements). For Digital interfaces: ▪ **[-1**] Not Configured = for ISDN spans, the progress indicator (PI) that is received in ISDN Proceeding, Progress, and Alerting messages is used as described in the options below. (default) ▪ **[0]** No PI = For IP-to-Tel calls, the device sends 180 Ringing SIP response to IP after receiving ISDN Alerting or (for CAS) after placing a call to PBX/PSTN. ▪ **[1]** PI =1, **[8]** PI =8: For IP-to-Tel calls, if the parameter EnableEarlyMedia is set to 1, the device sends 180 Ringing

Parameter	Description
	with SDP in response to an ISDN Alerting or it sends a 183 Session Progress message with SDP in response to only the first received ISDN Proceeding or Progress message after a call is placed to PBX/PSTN over the trunk.
[EnableRekeyAfter181]	Enables the device to send a Re-INVITE with a new (different) SRTP key (in the SDP) upon receipt of a SIP 181 response ("call is being forwarded"). ▪ **[0]** = Disable (default) ▪ **[1]** = Enable **Note:** This parameter is applicable only if SRTP is used.
[NumberOfActiveDialogs]	Defines the maximum number of active SIP dialogs that are not call related (i.e., REGISTER and SUBSCRIBE). This parameter is used to control the Registration/Subscription rate. The valid range is 1 to 20. The default value is 20.
[TransparentCoderOnDataCall[	▪ **[0]** = Only use coders from the coder list (default). ▪ **[1]** = Use Transparent coder for data calls (according to RFC 4040). The 'Transparent' coder can be used on data calls. When the device receives a Setup message from the ISDN with 'TransferCapabilities = data', it can initiate a call using the coder 'Transparent' (even if the coder is not included in the coder list). The initiated INVITE includes the following SDP attribute: a=rtpmap:97 CLEARMODE/8000 The default payload type is set according to the CodersGroup parameter. If the Transparent coder is not defined, the default value is set to 56. The payload type is negotiated with the remote side, i.e., the selected payload type is according to the remote side selection. The receiving device must include the 'Transparent' coder in its coder list.
Web/EMS: Default Release Cause **[DefaultReleaseCause]**	Defines the default Release Cause (sent to IP) for IP-to-Tel calls when the device initiates a call release and an explicit matching cause for this release is not found. The default release cause is NO_ROUTE_TO_DESTINATION (3). Other common values include NO_CIRCUIT_AVAILABLE (34), DESTINATION_OUT_OF_ORDER (27), etc. **Notes:** ▪ The default release cause is described in the Q.931 notation and is translated to corresponding SIP 40x or 50x values (e.g., 3 to SIP 404, and 34 to SIP 503). ▪ For analog interfaces: For an explanation on mapping PSTN release causes to SIP responses, refer to Mapping PSTN Release Cause to SIP Response on page 589. ▪ When the Trunk is disconnected or is not synchronized, the internal cause is 27. This cause is mapped, by default, to SIP 502.

Parameter	Description
	▪ For mapping SIP-to-Q.931 and Q.931-to-SIP release causes, refer to Configuring Release Cause Mapping on page 169. ▪ For a list of SIP responses-Q.931 release cause mapping, refer to "Release Reason Mapping" on page 626.
[IgnoreAlertAfterEarlyMedia]	Determines the device's interworking of Alerting messages from PRI to SIP. ▪ **[0]** = Disabled (default). ▪ **[1]** = Enabled. When enabled, if the device sends a 183 response with an SDP included and an Alerting message is then received from the Tel side (with or without Progress Indicator), the device does not send an additional 18x response and the voice channel remains open. When disabled, the device sends additional 18x responses as a result of receiving an Alerting message whether or not a 18x response was already sent.
Web: Enable Microsoft Extension **[EnableMicrosofExt]**	Modifies the called number for numbers received with Microsoft's proprietary "ext=xxx" parameter in the SIP INVITE URI user part. Microsoft Office Communications Server sometimes uses this proprietary parameter to indicate the extension number of the called party. ▪ **[0]** Disable (default). ▪ **[1]** Enable. For example, if a calling party makes a call to telephone number 622125519100 Ext. 104, the device receives the SIP INVITE (from Microsoft's application) with the URI user part as INVITE sip:622125519100;ext=104@10.1.1.10 (or INVITE tel:622125519100;ext=104). If the parameter EnableMicrosofExt is enabled, the device modifies the called number by adding an "e" as the prefix, removing the "ext=" parameter, and adding the extension number as the suffix (e.g., **e**622125519100**104**). Once modified, the device can then manipulate the number further, using the Number Manipulation tables (refer to "Number Manipulation and Routing Parameters" on page 428) to leave only the last 3 digits (for example) for sending to a PBX.
EMS: Use SIP URI For Diversion Header **[UseSIPURIForDiversionHeader]**	Defines the URI format in the SIP Diversion header. ▪ **[0]** = 'tel:' (default) ▪ **[1]** = 'sip:'
[TimeoutBetween100And18x]	Defines the timeout (in msec) between receiving a 100 Trying response and a subsequent 18x response. If a 18x response is not received before this timer expires, the call is disconnected. The valid range is 0 to 32,000. The default value is 0 (i.e., no timeout).
[EnableImmediateTrying]	Determines if and when the device sends a 100 Trying in response to an incoming INVITE request. ▪ **[0]** = 100 Trying response is sent upon receipt of a Proceeding message from the PSTN. ▪ **[1]** = 100 Trying response is sent immediately upon receipt of INVITE request (default).

Parameter	Description
[TransparentCoderPresentation]	Determines the format of the Transparent coder representation in the SDP. ▪ **[0]** = clearmode (default) ▪ **[1]** = X-CCD
Web: Comfort Noise Generation Negotiation EMS: Comfort Noise Generation **[ComfortNoiseNegotiation]**	Enables negotiation and usage of Comfort Noise (CN). ▪ **[0]** Disable (default). ▪ **[1]** Enable. The use of CN is indicated by including a payload type for CN on the media description line of the SDP. The device can use CN with a codec whose RTP time stamp clock rate is 8,000 Hz (G.711/G.726). The static payload type 13 is used. The use of CN is negotiated between sides. Therefore, if the remote side doesn't support CN, it is not used. **Note:** Silence Suppression must be enabled to generate CN.
Web/EMS: First Call Ringback Tone ID **[FirstCallRBTId]**	Determines the index of the first Ringback Tone in the CPT file. This option enables an Application server to request the device to play a distinctive Ringback tone to the calling party according to the destination of the call. The tone is played according to the Alert-Info header received in the 180 Ringing SIP response (the value of the Alert-Info header is added to the value of this parameter). The valid range is -1 to 1,000. The default value is -1 (i.e., play standard Ringback tone). **Notes:** ▪ It is assumed that all Ringback tones are defined in sequence in the CPT file. ▪ In case of an MLPP call, the device uses the value of this parameter plus 1 as the index of the Ringback tone in the CPT file (e.g., if this value is set to 1, then the index is 2, i.e., 1 + 1).
Web: Reanswer Time EMS: Regret Time **[RegretTime]**	For Analog interfaces: The time interval from when the user hangs up the phone until the call is disconnected (FXS). This allows the user to hang up and then pick up the phone (before this timeout) to continue the call conversation. Thus, it's also referred to as regret time. For Digital interfaces: Determines the time period the device waits for an MFC R2 Resume (Reanswer) signal once a Suspend (Clear back) signal is received from the PBX. If this timer expires, the call is released. Note that this is applicable only to the MFC-R2 CAS Brazil variant. The valid range is 0 to 255 (in seconds). The default value is 0.
Web: Enable Reanswering Info **[EnableReansweringINFO]**	Enables the device to send a SIP INFO message with the On-Hook/Off-Hook parameter when the FXS phone goes on-hook during an ongoing call and then off-hook again, within the user-defined regret timeout (configured by the parameter RegretTime). Therefore, the device notifies the far-end that the call has been re-answered. ▪ **[0]** Disable (default) ▪ **[1]** Enable

Parameter	Description
	This parameter is typically implemented for incoming IP-to-Tel collect calls to the FXS port. If the FXS user does not wish to accept the collect call, the user disconnects the call by on-hooking the phone. The device notifies the softswitch (or Application server) of the unanswered collect call (on-hook) by sending a SIP INFO message. As a result, the softswitch disconnects the call (sends a BYE message to the device). If the call is a regular incoming call and the FXS user on-hooks the phone without intending to disconnect the call, the softswitch does not disconnect the call (during the regret time). The INFO message format is as follows: INFO sip:12345@10.50.228.164:5082 SIP/2.0 Via: SIP/2.0/UDP 127.0.0.1;branch=z9hG4bK_05_905924040-90579 From: <sip:+551137077803@ims.acme.com.br:5080;user=phone>;tag=008277765 To: <sip:notavailable@unknown.invalid>;tag=svw-0-1229428367 Call-ID: ConorCCR-0-LU-1229417827103300@dtas-stdn.fs5000group0-000.l CSeq: 1 INFO Contact: sip:10.20.7.70:5060 Content-Type: application/On-Hook (application/Off-Hook) Content-Length: 0 **Notes:** ▪ This parameter is applicable only if the parameter RegretTime is configured. ▪ This parameter is applicable only to FXS interfaces.
Web: Enable IP2IP Application **[EnableIP2IPApplication]**	Enables the IP-to-IP Call Routing application. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Note:** For this parameter to take effect, a device reset is required.
Web/EMS: SIT Q850 Cause **[SITQ850Cause]**	Determines the Q.850 cause value specified in the SIP Reason header that is included in a 4xx response when a Special Information Tone (SIT) is detected on an IP-to-Tel call. The valid range is 0 to 127. The default value is 34. **Note:** For mapping specific SIT tones, you can use the SITQ850CauseForNC, SITQ850CauseForIC, SITQ850CauseForVC, and SITQ850CauseForRO parameters.
Web/EMS: SIT Q850 Cause For NC **[SITQ850CauseForNC]**	Determines the Q.850 cause value specified in the SIP Reason header that is included in a 4xx response when SIT-NC (No Circuit Found Special Information Tone) is detected from the PSTN for IP-to-Tel calls. The valid range is 0 to 127. The default value is 34. **Note:** When not configured (i.e., default), the SITQ850Cause parameter is used.

Parameter	Description
Web/EMS: SIT Q850 Cause For IC **[SITQ850CauseForIC]**	Determines the Q.850 cause value specified in the SIP Reason header that is included in a 4xx response when SIT-IC (Operator Intercept Special Information Tone) is detected from the PSTN for IP-to-Tel calls. The valid range is 0 to 127. The default value is -1 (not configured). **Note:** When not configured (i.e., default), the SITQ850Cause parameter is used.
Web/EMS: SIT Q850 Cause For VC **[SITQ850CauseForVC]**	Determines the Q.850 cause value specified in the SIP Reason header that is included in a 4xx response when SIT-VC (Vacant Circuit - non-registered number Special Information Tone) is detected from the PSTN for IP-to-Tel calls. The valid range is 0 to 127. The default value is -1 (not configured). **Note:** When not configured (i.e., default), the SITQ850Cause parameter is used.
Web/EMS: SIT Q850 Cause For RO **[SITQ850CauseForRO]**	Determines the Q.850 cause value specified in the SIP Reason header that is included in a 4xx response when SIT-RO (Reorder - System Busy Special Information Tone) is detected from the PSTN for IP-to-Tel calls. The valid range is 0 to 127. The default value is -1 (not configured). **Note:** When not configured (i.e., default), the SITQ850Cause parameter is used.
Out-of-Service (Busy Out) Parameters	
Web/EMS: Enable Busy Out **[EnableBusyOut]**	Determines whether the Busy Out feature is enabled. ▪ **[0]** Disable = 'Busy out' feature is not used (default). ▪ **[1]** Enable = 'Busy out' feature is enabled. When Busy Out is enabled and certain scenarios exist, the device performs the following: For analog interfaces: A reorder tone (configured by the parameter FXSOOSBehavior) is played when the phone is off-hooked. For digital interface: All E1/T1 trunks are automatically taken out of service by taking down the D-Channel or by sending a Service Out message for T1 PRI trunks supporting these messages (NI-2, 4/5-ESS, DMS-100, and Meridian). These behaviors are performed upon one of the following scenarios: ▪ Physically disconnected from the network (i.e., Ethernet cable is disconnected). ▪ The Ethernet cable is connected, but the device can't communicate with any host. Note that LAN Watch-Dog must be activated (the parameter EnableLANWatchDog set to 1). ▪ The device can't communicate with the proxy (according to the Proxy Keep-Alive mechanism) and no other alternative route exists to send the call.

Parameter	Description
	▪ The IP Connectivity mechanism is enabled (using the parameter AltRoutingTel2IPEnable) and there is no connectivity to any destination IP address. **Notes:** ▪ For Analog interfaces: The FXSOOSBehavior parameter determines the behavior of the FXS endpoints when a Busy Out or Graceful Lock occurs. ▪ For Analog interfaces: FXO endpoints during Busy Out and Lock are inactive. ▪ For Analog interfaces: Refer to the LifeLineType parameter for complementary optional behavior. ▪ For Digital interfaces: The Busy Out behavior varies between different protocol types. ▪ For Digital interfaces: The Busy-Out condition can also be applied to a specific Trunk Group. If there is no connectivity to the Serving IP Group of a specific Trunk Group (defined in the 'Trunk Group Settings' table), all the physical trunks pertaining to that Trunk Group are set to the Busy-Out condition. Each trunk uses the proper Out-Of-Service method according to the selected ISDN/CAS variant. ▪ For Digital interfaces: You can use the parameter DigitalOOSBehavior to select the method for setting digital trunks to Out-Of-Service.
Web: Out-Of-Service Behavior EMS:FXS OOS Behavior **[FXSOOSBehavior]**	Determines the behavior of undefined FXS endpoints and all FXS endpoints when a Busy Out condition exists. ▪ **[0]** None = Normal operation. No response is provided to undefined endpoints. A dial tone is played to FXS endpoints when a Busy Out condition exists. ▪ **[1]** Reorder Tone = The device plays a reorder tone to the connected phone/PBX (default). ▪ **[2]** Polarity Reversal = The device reverses the polarity of the endpoint marking it unusable (relevant, for example, for PBX DID lines). This option can't be configured on-the-fly. ▪ **[3]** Reorder Tone + Polarity Reversal = Same as 2 and 3 combined. This option can't be configured on-the-fly. ▪ **[4]** Current Disconnect = The device disconnects the current of the FXS endpoint. This option can't be configured on-the-fly. **Note:** This parameter is applicable only to FXS interfaces.
Retransmission Parameters	
Web: SIP T1 Retransmission Timer [msec] EMS: T1 RTX **[SipT1Rtx]**	The time interval (in msec) between the first transmission of a SIP message and the first retransmission of the same message. The default is 500. **Note:** The time interval between subsequent retransmissions of the same SIP message starts with SipT1Rtx and is multiplied by two until SipT2Rtx. For example (assuming that SipT1Rtx = 500 and SipT2Rtx = 4000): ▪ The first retransmission is sent after 500 msec. ▪ The second retransmission is sent after 1000 (2*500) msec.

Parameter	Description
	▪ The third retransmission is sent after 2000 (2*1000) msec. ▪ The fourth retransmission and subsequent retransmissions until SIPMaxRtx are sent after 4000 (2*2000) msec.
Web: SIP T2 Retransmission Timer [msec] EMS: T2 RTX **[SipT2Rtx]**	The maximum interval (in msec) between retransmissions of SIP messages. The default is 4000. **Note:** The time interval between subsequent retransmissions of the same SIP message starts with SipT1Rtx and is multiplied by two until SipT2Rtx.
Web: SIP Maximum RTX EMS: Max RTX **[SIPMaxRtx]**	Maximum number of UDP transmissions (first transmission plus retransmissions) of SIP messages. The range is 1 to 30. The default value is 7.
Web: Number of RTX Before Hot-Swap EMS: Proxy Hot Swap Rtx **[HotSwapRtx]**	Number of retransmitted INVITE/REGISTER messages before the call is routed (hot swap) to another Proxy/Registrar. The valid range is 1 to 30. The default value is 3. **Note:** This parameter is also used for alternative routing using the 'Outbound IP Routing Table'. If a domain name in the table is resolved into two IP addresses, and if there is no response for HotSwapRtx retransmissions to the INVITE message that is sent to the first IP address, the device immediately initiates a call to the second IP address.

6.7.2 Network Application Parameters

The SIP network application parameters are described in the table below:

Table 6-28: SIP Network Application Parameters

Parameter	Description
Web: Default CP Media Realm Name EMS: Default Realm Name **[cpDefaultMediaRealmName]**	For a description of this parameter, refer to "Configuring Media Realms" on page 99.
Web: SIP Media Realm Table **EMS: Protocol Definition > Media Realm**	
[CpMediaRealm]	This *ini* file table parameter configures the SIP Media Realm table. The Media Realm table allows you to divide a Media-type interface (defined in the 'Multiple Interface' table) into several realms, where each realm is specified by a UDP port range. The format of this parameter is as follows: [CpMediaRealm] FORMAT CpMediaRealm_Index = CpMediaRealm_MediaRealmName, CpMediaRealm_IPv4IF, CpMediaRealm_IPv6IF, CpMediaRealm_PortRangeStart, CpMediaRealm_MediaSessionLeg, CpMediaRealm_PortRangeEnd; [\CpMediaRealm] For example, CpMediaRealm 1 = Mrealm1, Voice, , 6600, 20, 6790; CpMediaRealm 2 = Mrealm2, Voice, , 6800, 10, 6890; **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ This table can include up to 16 indices (where 0 is the first index). However, only up to 8 media realms can be used by the device (as a maximum of 8 IP Groups can be configured). ▪ Each table index must be unique. ▪ The parameter cpDefaultRealmName can be used to define one of the Media Realms appearing in this table as the default Media Realm. If the parameter cpDefaultRealmName is not configured, then the first Media Realm appearing in this table is set as default. If this table is not configured, then the default Media Realm includes all defined media interfaces. ▪ A Media Realm can be assigned to an IP Group (in the 'IP Group' table) or an SRD (in the 'SRD' table). If different Media Realms are assigned to both an IP Group and SRD, the IP Group's Media Realm takes precedence. ▪ The parameter IPv6IF is not applicable. ▪ For a detailed description of all the parameters included in this *ini* file table parameter and for configuring Media Realms using the Web interface, refer to "Configuring Media Realms" on page 99. ▪ For a description on configuring *ini* file table parameters, refer to "Configuring *ini* File Table Parameters" on page 222.

Parameter	Description
Web :Signaling Routing Domain (SRD) Table **EMS: SRD Table**	
[SRD]	This *ini* file table parameter configures the Signaling Routing Domain (SRD) table. The format of this parameter is as follows: [SRD] FORMAT SRD_Index = SRD_Name, SRD_MediaRealm; [\SRD] For example: SRD 1 = LAN_SRD, Mrealm1; SRD 2 = WAN_SRD, Mrealm2; **Notes:** ▪ This table can include up to 5 indices (where 0 is the first index). ▪ For a detailed description of the table's individual parameters and for configuring the table using the Web interface, refer to "Configuring the Signaling Routing Domain Table" on page 110. ▪ For an explanation on using *ini* file table parameters, refer to "Configuring *ini* File Table Parameters" on page 222.
Web: SIP Interface Table **EMS: SIP Interfaces**	
[SIPInterface]	This *ini* file table parameter configures the SIP Interface table. The SIP Interface represents a SIP signaling entity, comprising ports (UDP, TCP, and TLS) and associated with a specific IP address and an SRD ID. The format of this parameter is as follows: [SIPInterface] FORMAT SIPInterface_Index = SIPInterface_NetworkInterface, SIPInterface_ApplicationType, SIPInterface_UDPPort, SIPInterface_TCPPort, SIPInterface_TLSPort, SIPInterface_SRD; [\SIPInterface] For example: SIPInterface 0 = Media1, 2, 5060, 5060, 5061, 1; SIPInterface 1 = Media1, 2, 5070, 5070, 5071, 2; SIPInterface 2 = Voice, 0, 5090, 5000, 5081, 2; **Notes:** ▪ This table can include up to 6 indices (where 0 is the first index). ▪ Each SIP Interface must have a unique signaling port (i.e., no two SIP Interfaces can share the same port - no port overlapping). ▪ You can define up to three different SIP Interfaces per SRD, where each SIP Interface pertains to a different application type (i.e., GW, SAS). ▪ For a detailed description of the table's individual parameters and for configuring the table using the Web interface, refer to "Configuring the SIP Interface Table" on page 112. ▪ For an explanation on using *ini* file table parameters, refer to "Configuring *ini* File Table Parameters" on page 222.

6.7.3 IP Group, Proxy, Registration and Authentication Parameters

The proxy server, registration and authentication SIP parameters are described in the table below.

Table 6-29: Proxy, Registration and Authentication SIP Parameters

Parameter	Description
Web: IP Group Table **EMS: Endpoints > IP Group**	
[IPGroup]	This *ini* file table parameter configures the IP Group table. The format of this parameter is as follows: [IPGroup] FORMAT IPGroup_Index = IPGroup_Type, IPGroup_Description, IPGroup_ProxySetId, IPGroup_SIPGroupName, IPGroup_ContactUser, IPGroup_EnableSurvivability, IPGroup_ServingIPGroup, IPGroup_SipReRoutingMode, IPGroup_AlwaysUseRouteTable, IPGroup_RoutingMode, IPGroup_SRD, IPGroup_MediaRealm, IPGroup_ClassifyByProxySet, IPGroup_ProfileId; [\IPGroup] For example: IPGroup 1 = 0, "dol gateway", 1, firstIPgroup, , 0, -1, 0, 0, -1, 0, mrealm1, 1, 1; IPGroup 2 = 0, "abc server", 2, secondIPgroup, , 0, -1, 0, 0, -1, 0, mrealm2, 1, 2; IPGroup 3 = 1, "IP phones", 1, thirdIPGroup, , 0, -1, 0, 0, -1, 0, mrealm3, 1, 2; **Notes:** ▪ This table parameter can include up to 9 indices (1-9). ▪ The parameters Type, RoutingMode, EnableSurvivability, ServingIPGroup, SRD, and ClassifyByProxySet are not applicable to Mediant 600. ▪ For a detailed description of the *ini* file table's parameters and for configuring this table using the Web interface, refer to "Configuring the IP Groups" on page 113. ▪ For an explanation on using *ini* file table parameters, refer to "Configuring *ini* File Table Parameters" on page 222.

Parameter	Description
Web: Authentication Table **EMS: SIP Endpoints > Authentication**	
[Authentication]	This *ini* file table parameter defines a user name and password for authenticating each device port. The format of this parameter is as follows: [Authentication] FORMAT Authentication_Index = Authentication_UserId, Authentication_UserPassword, Authentication_Module, Authentication_Port; [\Authentication] Where, ▪ UserId = User name ▪ UserPassword = Password ▪ Module = Module number (where 1 depicts the module in Slot 1) ▪ Port = Port number (where 1 depicts the Port 1 of the module) For example: [Authentication] Authentication 0 = john,1325,1,1; (user name "john" with password 1325 for authenticating Port 1 of Module 1) Authentication 1 = lee,1552,1,2; (user name "lee" with password 1552 for authenticating Port 2 of Module 1) [\Authentication] **Notes:** ▪ The indexing of this parameter starts at 0. ▪ The parameter AuthenticationMode determines whether authentication is performed per port or for the entire device. If authentication is performed for the entire device, the configuration in this table parameter is ignored. ▪ If the user name or password are not configured, the port's phone number (configured using the parameter TrunkGroup - Trunk Group Table) and global password (using the individual parameter Password) are used for authentication. ▪ Authentication is typically used for FXS interfaces, but can also be used for FXO interfaces. ▪ For configuring the Authentication table using the Web interface, refer to Configuring Authentication on page 171. ▪ For an explanation on using *ini* file table parameters, refer to "Configuring *ini* File Table Parameters" on page 222.

Parameter	Description
Web: Account Table **EMS: SIP Endpoints > Account**	
[Account]	This *ini* file table parameter configures the Account table for registering and/or authenticating (digest) Trunk Groups or IP Groups (e.g., an IP-PBX) to a Serving IP Group (e.g., an Internet Telephony Service Provider - ITSP). The format of this parameter is as follows: [Account] FORMAT Account_Index = Account_ServedTrunkGroup, Account_ServedIPGroup, Account_ServingIPGroup, Account_Username, Account_Password, Account_HostName, Account_Register, Account_ContactUser, Account_ApplicationType; [\Account] For example: Account 1 = 1, -1, 1, user, 1234, acl, 1, ITSP1; **Notes:** ▪ This table can include up to 10 indices (where 1 is the first index). ▪ The parameter Account_ApplicationType is not applicable. ▪ You can define multiple table indices with the same ServedTrunkGroup but different ServingIPGroups, username, password, HostName, and ContactUser. This provides the capability for registering the same Trunk Group or IP Group to several ITSP's (i.e., Serving IP Groups). ▪ For a detailed description of this table's parameters and for configuring this table using the Web interface, refer to "Configuring the Account Table" on page 118. ▪ For an explanation on using *ini* file table parameters, refer to "Configuring *ini* File Table Parameters" on page 222.
Proxy Registration Parameters	
Web: Use Default Proxy EMS: Proxy Used **[IsProxyUsed]**	Enables the use of a SIP Proxy server. ▪ **[0]** No = Proxy isn't used and instead, the internal routing table is used (default). ▪ **[1]** Yes = Proxy is used. If you are using a Proxy server, enter the IP address of the Proxy server in the 'Proxy Sets table' (refer to "Configuring the Proxy Sets Table" on page 123). If you are not using a Proxy server, you must configure the 'Outbound IP Routing Table' (described in "Configuring the Outbound IP Routing Table" on page 158).
Web/EMS: Proxy Name **[ProxyName]**	Defines the Home Proxy domain name. If specified, this name is used as the Request-URI in REGISTER, INVITE, and other SIP messages, and as the host part of the To header in INVITE messages. If not specified, the Proxy IP address is used instead. The value must be string of up to 49 characters.

Parameter	Description
Web: Redundancy Mode EMS: Proxy Redundancy Mode **[ProxyRedundancyMode]**	Determines whether the device switches back to the primary Proxy after using a redundant Proxy. ▪ **[0]** Parking = device continues working with a redundant (now active) Proxy until the next failure, after which it works with the next redundant Proxy (default). ▪ **[1]** Homing = device always tries to work with the primary Proxy server (i.e., switches back to the primary Proxy whenever it's available). **Note:** To use this Proxy Redundancy mechanism, you need to enable the keep-alive with Proxy option, by setting the parameter EnableProxyKeepAlive to 1 or 2.
Web: Proxy IP List Refresh Time EMS: IP List Refresh Time **[ProxyIPListRefreshTime]**	Defines the time interval (in seconds) between each Proxy IP list refresh. The range is 5 to 2,000,000. The default interval is 60.
Web: Enable Fallback to Routing Table EMS: Fallback Used **[IsFallbackUsed]**	Determines whether the device falls back to the 'Outbound IP Routing Table' for call routing when Proxy servers are unavailable. ▪ **[0]** Disable = Fallback is not used (default). ▪ **[1]** Enable = The 'Outbound IP Routing Table' is used when Proxy servers are unavailable. When the device falls back to the 'Outbound IP Routing Table', it continues scanning for a Proxy. When the device locates an active Proxy, it switches from internal routing back to Proxy routing. **Note:** To enable the redundant Proxies mechanism, set the parameter EnableProxyKeepAlive to 1 or 2.
Web/EMS: Prefer Routing Table **[PreferRouteTable]**	Determines whether the device's internal routing table takes precedence over a Proxy for routing calls. ▪ **[0]** No = Only a Proxy server is used to route calls (default). ▪ **[1]** Yes = The device checks the routing rules in the 'Outbound IP Routing Table' for a match with the Tel-to-IP call. Only if a match is not found is a Proxy used.
Web/EMS: Always Use Proxy **[AlwaysSendToProxy]**	Determines whether the device sends SIP messages and responses through a Proxy server. ▪ **[0]** Disable = Use standard SIP routing rules (default). ▪ **[1]** Enable = All SIP messages and responses are sent to the Proxy server. **Note:** This parameter is applicable only if a Proxy server is used (i.e., the parameter IsProxyUsed is set to 1).
Web: SIP ReRouting Mode EMS: SIP Re-Routing Mode **[SIPReroutingMode]**	Determines the routing mode after a call redirection (i.e., a 3xx SIP response is received) or transfer (i.e., a SIP REFER request is received). ▪ **[0]** Standard = INVITE messages that are generated as a result of Transfer or Redirect are sent directly to the URI, according to the Refer-To header in the REFER message, or Contact header in the 3xx response (default). ▪ **[1]** Proxy = Sends a new INVITE to the Proxy.

Parameter	Description
	Note: This option is applicable only if a Proxy server is used and the parameter AlwaysSendtoProxy is set to 0. ▪ **[2]** Routing Table = Uses the Routing table to locate the destination and then sends a new INVITE to this destination. **Notes:** ▪ When this parameter is set to **[1]** and the INVITE sent to the Proxy fails, the device re-routes the call according to the Standard mode **[0].** ▪ When this parameter is set to **[2]** and the INVITE fails, the device re-routes the call according to the Standard mode **[0].** If DNS resolution fails, the device attempts to route the call to the Proxy. If routing to the Proxy also fails, the Redirect/Transfer request is rejected. ▪ When this parameter is set to **[2],** the XferPrefix parameter can be used to define different routing rules for redirect calls. ▪ This parameter is disregarded if the parameter AlwaysSendToProxy is set to 1.
Web/EMS: DNS Query Type **[DNSQueryType]**	Enables the use of DNS Naming Authority Pointer (NAPTR) and Service Record (SRV) queries to resolve Proxy and Registrar servers and to resolve all domain names that appear in the SIP Contact and Record-Route headers. ▪ **[0]** A-Record (default) ▪ **[1]** SRV ▪ **[2]** NAPTR If set to A-Record **[0],** no NAPTR or SRV queries are performed. If set to SRV **[1]** and the Proxy/Registrar IP address parameter, Contact/Record-Route headers, or IP address defined in the Routing tables contain a domain name, an SRV query is performed. The device uses the first host name received from the SRV query. The device then performs a DNS A-record query for the host name to locate an IP address. If set to NAPTR **[2],** an NAPTR query is performed. If it is successful, an SRV query is sent according to the information received in the NAPTR response. If the NAPTR query fails, an SRV query is performed according to the configured transport type. If the Proxy/Registrar IP address parameter, the domain name in the Contact/Record-Route headers, or the IP address defined in the Routing tables contain a domain name with port definition, the device performs a regular DNS A-record query. If a specific Transport Type is defined, a NAPTR query is not performed. **Note:** To enable NAPTR/SRV queries for Proxy servers only, use the parameter ProxyDNSQueryType.

Parameter	Description
Web: Proxy DNS Query Type **[ProxyDNSQueryType]**	Enables the use of DNS Naming Authority Pointer (NAPTR) and Service Record (SRV) queries to discover Proxy servers. ▪ **[0]** A-Record (default) ▪ **[1]** SRV ▪ **[2]** NAPTR If set to A-Record **[0],** no NAPTR or SRV queries are performed. If set to SRV **[1]** and the Proxy IP address parameter contains a domain name without port definition (e.g., ProxyIP = domain.com), an SRV query is performed. The SRV query returns up to four Proxy host names and their weights. The device then performs DNS A-record queries for each Proxy host name (according to the received weights) to locate up to four Proxy IP addresses. Therefore, if the first SRV query returns two domain names and the A-record queries return two IP addresses each, no additional searches are performed. If set to NAPTR **[2],** an NAPTR query is performed. If it is successful, an SRV query is sent according to the information received in the NAPTR response. If the NAPTR query fails, an SRV query is performed according to the configured transport type. If the Proxy IP address parameter contains a domain name with port definition (e.g., ProxyIP = domain.com:5080), the device performs a regular DNS A-record query. If a specific Transport Type is defined, a NAPTR query is not performed. **Note:** When enabled, NAPTR/SRV queries are used to discover Proxy servers even if the parameter DNSQueryType is disabled.
Web/EMS: Graceful Busy Out Timeout [sec] **[GracefulBusyOutTimeout]**	Determines the timeout interval (in seconds) for Out of Service (OOS) graceful shutdown mode for busy trunks (per trunk) if communication fails with a Proxy server (or Proxy Set). In such a scenario, the device rejects new calls from the PSTN (Serving Trunk Group), but maintains currently active calls for this user-defined timeout. Once this timeout elapses, the device terminates currently active calls and takes the trunk out of service (sending the PSTN busy-out signal). Trunks on which no calls are active are immediately taken out of service regardless of the timeout. The range is 0 to 3,600. The default is 0. **Note:** This parameter is applicable only to digital interfaces.
Web/EMS: Use Gateway Name for OPTIONS **[UseGatewayNameForOptions]**	Determines whether the device uses its IP address or gateway name in keep-alive SIP OPTIONS messages. ▪ **[0]** No = Use the device's IP address in keep-alive OPTIONS messages (default). ▪ **[1]** Yes = Use 'Gateway Name' (SIPGatewayName) in keep-alive OPTIONS messages. The OPTIONS Request-URI host part contains either the device's IP address or a string defined by the parameter

Parameter	Description
	SIPGatewayName. The device uses the OPTIONS request as a keep-alive message to its primary and redundant Proxies (i.e., the parameter EnableProxyKeepAlive is set to 1).
Web/EMS: User Name **[UserName]**	User name used for Registration and Basic/Digest authentication with a Proxy/Registrar server. The default value is an empty string. **Notes:** ▪ This parameter is applicable only if single device registration is used (i.e., the parameter AuthenticationMode is set to authentication per gateway). ▪ Instead of configuring this parameter, the Authentication table can be used (refer to Authentication on page 171).
Web/EMS: Password **[Password]**	The password used for Basic/Digest authentication with a Proxy/Registrar server. A single password is used for all device ports. The default is 'Default_Passwd'. **Note:** Instead of configuring this parameter, the Authentication table can be used (refer to Authentication on page 171).
Web/EMS: Cnonce **[Cnonce]**	Cnonce string used by the SIP server and client to provide mutual authentication. The value is free format, i.e., 'Cnonce = 0a4f113b'. The default is 'Default_Cnonce'.
Web/EMS: Mutual Authentication Mode **[MutualAuthenticationMode]**	Determines the device's mode of operation when Authentication and Key Agreement (AKA) Digest Authentication is used. ▪ **[0]** Optional = Incoming requests that don't include AKA authentication information are accepted (default). ▪ **[1]** Mandatory = Incoming requests that don't include AKA authentication information are rejected.
Web/EMS: Challenge Caching Mode **[SIPChallengeCachingMode]**	Determines the mode for Challenge Caching, which reduces the number of SIP messages transmitted through the network. The first request to the Proxy is sent without authorization. The Proxy sends a 401/407 response with a challenge. This response is saved for further uses. A new request is re-sent with the appropriate credentials. Subsequent requests to the Proxy are automatically sent with credentials (calculated from the saved challenge). If the Proxy doesn't accept the new request and sends another challenge, the old challenge is replaced with the new one. ▪ **[0]** None = Challenges are not cached. Every new request is sent without preliminary authorization. If the request is challenged, a new request with authorization data is sent. (default) ▪ **[1]** INVITE Only = Challenges issued for INVITE requests are cached. This prevents a mixture of REGISTER and INVITE authorizations. ▪ **[2]** Full = Caches all challenges from the proxies. **Note**: Challenge Caching is used with all proxies and not only with the active one.

Parameter	Description
Web: Proxy IP Table **EMS: Proxy IP**	
[ProxyIP]	This *ini* file table parameter configures the Proxy Set table with up to six Proxy Set IDs, each with up to five Proxy server IP addresses (or fully qualified domain name/FQDN). Each Proxy Set can be defined with a transport type (UDP, TCP, or TLS). The format of this parameter is as follows: [ProxyIP] FORMAT ProxyIp_Index = ProxyIp_IpAddress, ProxyIp_TransportType, ProxyIp_ProxySetId; [\ProxyIP] For example: ProxyIp 0 = 10.33.37.77, -1, 0; ProxyIp 1 = 10.8.8.10, 0, 2; ProxyIp 2 = 10.5.6.7, -1, 1; **Notes:** ▪ This parameter can include up to 30 indices (0-29). ▪ The Proxy Set represents the destination of the call. ▪ For assigning various attributes (such as Proxy Load Balancing) per Proxy Set ID, use the parameter ProxySet. ▪ For configuring the Proxy Set ID table using the Web interface and for a detailed description of the parameters of this *ini* file table, refer to "Configuring the Proxy Sets Table" on page 123. ▪ For an explanation on using *ini* file table parameters, refer to "Configuring *ini* File Table Parameters" on page 222.
Web: Proxy Set Table **EMS: Proxy Set**	
[ProxySet]	This *ini* file table parameter configures the Proxy Set ID table. It is used in conjunction with the *ini* file table parameter ProxyIP, which defines the Proxy Set IDs with their IP addresses. The ProxySet *ini* file table parameter defines additional attributes per Proxy Set ID. This includes, for example, Proxy keep-alive and load balancing and redundancy mechanisms (if a Proxy Set contains more than one proxy address). The format of this parameter is as follows: [ProxySet] FORMAT ProxySet_Index = ProxySet_EnableProxyKeepAlive, ProxySet_ProxyKeepAliveTime, ProxySet_ProxyLoadBalancingMethod, ProxySet_IsProxyHotSwap, ProxySet_SRD; [\ProxySet] For example: ProxySet 0 = 0, 60, 0, 0, 0; ProxySet 1 = 1, 60, 1, 0, 1; **Notes:** ▪ This table parameter can include up to 10 indices (0-9).

Parameter	Description
	▪ For configuring the Proxy Set IDs and their IP addresses, use the parameter ProxyIP. ▪ For configuring the Proxy Set ID table using the Web interface and for a detailed description of the parameters of this *ini* file table, refer to "Configuring the Proxy Sets Table" on page 123. ▪ For an explanation on using *ini* file table parameters, refer to "Configuring *ini* File Table Parameters" on page 222.
Registrar Parameters	
Web: Enable Registration EMS: Is Register Needed **[IsRegisterNeeded]**	Enables the device to register to a Proxy/Registrar server. ▪ **[0]** Disable = The device doesn't register to Proxy/Registrar server (default). ▪ **[1]** Enable = The device registers to Proxy/Registrar server when the device is powered up and at every user-defined interval (configured by the parameter RegistrationTime). **Note:** The device sends a REGISTER request for each channel or for the entire device (according to the AuthenticationMode parameter).
Web/EMS: Registrar Name **[RegistrarName]**	Registrar domain name. If specified, the name is used as the Request-URI in REGISTER messages. If it isn't specified (default), the Registrar IP address, or Proxy name or IP address is used instead. The valid range is up to 49 characters.
Web: Registrar IP Address EMS: Registrar IP **[RegistrarIP]**	The IP address (or FQDN) and port number (optional) of the Registrar server. The IP address is in dotted-decimal notation, e.g., 201.10.8.1:<5080>. **Notes:** ▪ If not specified, the REGISTER request is sent to the primary Proxy server. ▪ When a port number is specified, DNS NAPTR/SRV queries aren't performed, even if the parameter DNSQueryType is set to 1 or 2. ▪ If the parameter RegistrarIP is set to an FQDN and is resolved to multiple addresses, the device also provides real-time switching (hotswap mode) between different Registrar IP addresses (the parameter IsProxyHotSwap is set to 1). If the first Registrar doesn't respond to the REGISTER message, the same REGISTER message is sent immediately to the next Proxy. To allow this mechanism, the parameter EnableProxyKeepAlive must be set to 0. ▪ When a specific transport type is defined using the parameter RegistrarTransportType, a DNS NAPTR query is not performed even if the parameter DNSQueryType is set to 2.

Parameter	Description
Web/EMS: Registrar Transport Type **[RegistrarTransportType]**	Determines the transport layer used for outgoing SIP dialogs initiated by the device to the Registrar. ▪ **[-1]** Not Configured (default) ▪ **[0]** UDP ▪ **[1]** TCP ▪ **[2]** TLS **Note:** When set to 'Not Configured', the value of the parameter SIPTransportType is used.
Web/EMS: Registration Time **[RegistrationTime]**	Defines the time interval (in seconds) for registering to a Proxy server. The value is used in the SIP Expires header. In addition, this parameter defines the time interval between Keep-Alive messages when the parameter EnableProxyKeepAlive is set to 2 (REGISTER). Typically, the device registers every 3,600 sec (i.e., one hour). The device resumes registration according to the parameter RegistrationTimeDivider. The valid range is 10 to 2,000,000. The default value is 180.
Web: Re-registration Timing **[%]** EMS: Time Divider **[RegistrationTimeDivider]**	Defines the re-registration timing (in percentage). The timing is a percentage of the re-register timing set by the Registrar server. The valid range is 50 to 100. The default value is 50. For example: If this parameter is set to 70% and the Registration Expires time is 3600, the device re-sends its registration request after 3600 x 70% (i.e., 2520 sec). **Note**: This parameter may be overridden if the parameter RegistrationTimeThreshold is greater than 0.
Web/EMS: Registration Retry Time **[RegistrationRetryTime]**	Defines the time interval (in seconds) after which a registration request is re-sent if registration fails with a 4xx response or if there is no response from the Proxy/Registrar server. The default is 30 seconds. The range is 10 to 3600.
Web: Registration Time Threshold EMS: Time Threshold **[RegistrationTimeThreshold]**	Defines a threshold (in seconds) for re-registration timing. If this parameter is greater than 0, but lower than the computed re-registration timing (according to the parameter RegistrationTimeDivider), the re-registration timing is set to the following: timing set by the Registration server in the SIP Expires header minus the value of the parameter RegistrationTimeThreshold. The valid range is 0 to 2,000,000. The default value is 0.
Web: Re-register On INVITE Failure EMS: Register On Invite Failure **[RegisterOnInviteFailure]**	Enables immediate re-registration if a failure response is received for an INVITE request sent by the device. ▪ **[0]** Disable (default) ▪ **[1]** Enable
Web: ReRegister On Connection Failure EMS: Re Register On Connection Failure **[ReRegisterOnConnectionFailure]**	Enables the device to perform SIP re-registration upon TCP/TLS connection failure. ▪ **[0]** Disable (default) ▪ **[1]** Enable

Parameter	Description
Web: Gateway Registration Name EMS: Name **[GWRegistrationName]**	Defines the user name that is used in the From and To headers in SIP REGISTER messages. If no value is specified (default) for this parameter, the UserName parameter is used instead. **Note:** This parameter is applicable only for single registration per device (i.e., AuthenticationMode is set to 1). When the device registers each channel separately (i.e., AuthenticationMode is set to 0), the user name is set to the channel's phone number.
Web/EMS: Authentication Mode **[AuthenticationMode]**	Determines the device's registration and authentication method. ▪ **[0]** Per Endpoint = Registration and authentication is performed separately for each endpoint/B-channel. ▪ **[1]** Per Gateway = Single registration and authentication for the entire device (default). ▪ **[3]** Per FXS = Registration and authentication for FXS endpoints. Typically, authentication per endpoint is used for FXS interfaces,where each endpoint registers (and authenticates) separately with its own user name and password. Single registration and authentication (Authentication Mode = 1) is usually defined for FXO and digital modules.
Web: Set Out-Of-Service On Registration Failure EMS: Set OOS On Registration Fail **[OOSOnRegistrationFail]**	Enables setting an endpoint, trunk, or the entire device (i.e., all endpoints) to out-of-service if registration fails. ▪ **[0]** Disable (default) ▪ **[1]** Enable If the registration is per endpoint (i.e., AuthenticationMode is set to 0) or per Account (refer to "Configuring Trunk Group Settings" on page 104) and a specific endpoint/Account registration fails (SIP 4xx or no response), then that endpoint is set to out-of-service until a success response is received in a subsequent registration request. When the registration is per the entire device (i.e., AuthenticationMode is set to 1) and registration fails, all endpoints are set to out-of-service. If all the Accounts of a specific Trunk Group fail registration and if the Trunk Group comprises a complete trunk, then the entire trunk is set to out-of-service. **Note:** Te out-of-service method is configured using the parameter FXSOOSBehavior.
[UnregistrationMode]	Determines whether the device performs an explicit unregister. ▪ **[0]** Disable (default) ▪ **[1]** Enable = The device sends an asterisk ("*") value in the SIP Contact header, instructing the Registrar server to remove all previous registration bindings. When enabled, the device removes SIP User Agent (UA) registration bindings in a Registrar, according to RFC 3261. Registrations are soft state and expire unless refreshed, but they can also be explicitly removed. A client can attempt to influence the expiration interval selected by the Registrar. A UA requests the immediate removal of a binding by specifying an expiration interval of "0" for that contact address in a

Parameter	Description
	REGISTER request. UA's should support this mechanism so that bindings can be removed before their expiration interval has passed. Use of the "*" Contact header field value allows a registering UA to remove all bindings associated with an address-of-record (AOR) without knowing their precise values. **Note:** The REGISTER-specific Contact header field value of "*" applies to all registrations, but it can only be used if the Expires header field is present with a value of "0".

6.7.4 Voice Mail Parameters

The voice mail parameters are described in the table below. For detailed information on the Voice Mail application, refer to the *CPE Configuration Guide for Voice Mail*.

Table 6-30: Voice Mail Parameters

Parameter	Description
Web/EMS: Voice Mail Interface **[VoiceMailInterface]**	Enables the device's Voice Mail application and determines the communication method used between the PBX and the device. ▪ **[0]** None (default) ▪ **[1]** DTMF ▪ **[2]** SMDI ▪ **[3]** QSIG ▪ **[4]** SETUP Only (ISDN) ▪ **[5]** MATRA/AASTRA QSIG ▪ **[6]** QSIG SIEMENS = QSIG MWI activate and deactivate messages include Siemens Manufacturer Specific Information (MSI) ▪ **[7]** IP2IP = The device's IP2IP application is used for interworking between an IP Voice Mail server and the device. This is implemented for sending unsolicited SIP NOTIFY messages received from the Voice Mail server to an IP Group (configured using the parameter NotificationIPGroupID). **Note:** To enable voice mail per Trunk Group, you can use a Tel Profile ID that is configured with voice mail interface enabled. This eliminates the phenomenon of call delay on Trunks not implementing voice mail when voice mail is enabled using this global parameter.

Parameter	Description
Web: Enable VoiceMail URI EMS: Enable VMURI **[EnableVMURI]**	Enables or disables the interworking of target and cause for redirection from Tel to IP and vice versa, according to RFC 4468. ▪ **[0]** Disable = Disable (default). ▪ **[1]** Enable = Enable Upon receipt of an ISDN Setup message with Redirect values, the device maps the Redirect phone number to the SIP 'target' parameter and the Redirect number reason to the SIP 'cause' parameter in the Request-URI. **Redirecting Reason >> SIP Response Code** Unknown >> 404 User busy >> 486 No reply >> 408 Deflection >> 487/480 Unconditional >> 302 Others >> 302 If the device receives a Request-URI that includes a 'target' and 'cause' parameter, the 'target' is mapped to the Redirect phone number and the 'cause' is mapped to the Redirect number reason.
Web/EMS: Line Transfer Mode **[LineTransferMode]**	Determines the call transfer method used by the device. ▪ **[0]** None = IP (default). ▪ **[1]** Blind = PBX blind transfer. After receiving a REFER message from the IP side, the FXO sends a hook-flash to the PBX, dials the digits (that are received in the Refer-To header), and then immediately drops the line (on-hook). The PBX performs the transfer internally. ▪ **[2]** Semi Supervised = PBX Semi-Supervised transfer. After receiving a REFER message from the IP side, the FXO sends a hook-flash to the PBX, and then dials the digits (that are received in the Refer-To header). If no Busy or Reorder tones are detected (within approximately 2 seconds), the device completes the call transfer by releasing the line; otherwise, the transfer is cancelled, the device sends a SIP NOTIFY message with a failure reason in the NOTIFY body (such as 486 if busy tone detected), and generates an additional hook-flash towards the FXO line to restore connection to the original call. ▪ **[3]** Supervised = PBX Supervised transfer. After receiving a REFER message from the IP side, the FXO sends a hook-flash to the PBX, and then dials the digits (that are received in the Refer-To header). The FXO waits for connection of the transfer call and if speech is detected (e.g., "hello") within approximately 2 seconds, the device completes the call transfer by releasing the line; otherwise, the transfer is cancelled, the device sends a SIP NOTIFY message with a failure reason in the NOTIFY body (such as 486 if busy tone detected) and generates an additional hook-flash towards the FXO line to restore connection to the original call.

Parameter	Description
SMDI Parameters	
Web/EMS: Enable SMDI **[SMDI]**	Enables Simplified Message Desk Interface (SMDI) interface on the device. ▪ **[0]** Disable = Normal serial (default) ▪ **[1]** Enable (Bellcore) ▪ **[2]** Ericsson MD-110 ▪ **[3]** NEC (ICS) **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ When the RS-232 connection is used for SMDI messages (Serial SMDI), it cannot be used for other applications, for example, to access the Command Line Interface (CLI).
Web/EMS: SMDI Timeout **[SMDITimeOut]**	Determines the time (in msec) that the device waits for an SMDI Call Status message before or after a Setup message is received. This parameter synchronizes the SMDI and analog CAS interfaces. If the timeout expires and only an SMDI message is received, the SMDI message is dropped. If the timeout expires and only a Setup message is received, the call is established. The valid range is 0 to 10000 (i.e., 10 seconds). The default value is 2000.
Message Waiting Indication (MWI) Parameters	
Web: MWI Off Digit Pattern EMS: MWI Off Code **[MWIOffCode]**	Determines the digit code used by the device to notify the PBX that there aren't any messages waiting for a specific extension. This code is added as prefix to the dialed number. The valid range is a 25-character string.
Web: MWI On Digit Pattern EMS: MWI On Code **[MWIOnCode]**	Determines the digit code used by the device to notify the PBX of messages waiting for a specific extension. This code is added as prefix to the dialed number. The valid range is a 25-character string.
Web: MWI Suffix Pattern EMS: MWI Suffix Code **[MWISuffixCode]**	Determines the digit code used by the device as a suffix for 'MWI On Digit Pattern' and 'MWI Off Digit Pattern'. This suffix is added to the generated DTMF string after the extension number. The valid range is a 25-character string.
Web: MWI Source Number EMS: MWI Source Name **[MWISourceNumber]**	Determines the calling party's phone number used in the Q.931 MWI Setup message to PSTN. If not configured, the channel's phone number is used as the calling number.
[MWIQsigMsgCentreIdIDParty Number]	Defines the Message Centred ID party number used for QSIG MWI messages. If not configured (default), the parameter is not included in MWI QSIG messages.
[NotificationIPGroupID]	Determines the IP Group ID to which the device sends SIP NOTIFY MWI messages (for MWI Interrogation). **Notes:** ▪ For a detailed description on the interworking of QSIG MWI to IP, refer to Message Waiting Indication on page 572. ▪ To determine the handling method for MWI Interrogation messages, use the MWIInterrogationType parameter.

Parameter	Description
Digit Patterns The following digit pattern parameters apply only to voice mail applications that use the DTMF communication method. For the available pattern syntaxes, refer to the *CPE Configuration Guide for Voice Mail*.	
Web: Forward on Busy Digit Pattern (Internal) EMS: Digit Pattern Forward On Busy **[DigitPatternForwardOnBusy]**	Determines the digit pattern used by the PBX to indicate 'call forward on busy' when the original call is received from an internal extension. The valid range is a 120-character string.
Web: Forward on No Answer Digit Pattern (Internal) EMS: Digit Pattern Forward On No Answer **[DigitPatternForwardOnNoAnswer]**	Determines the digit pattern used by the PBX to indicate 'call forward on no answer' when the original call is received from an internal extension. The valid range is a 120-character string.
Web: Forward on Do Not Disturb Digit Pattern (Internal) EMS: Digit Pattern Forward On DND **[DigitPatternForwardOnDND]**	Determines the digit pattern used by the PBX to indicate 'call forward on do not disturb' when the original call is received from an internal extension. The valid range is a 120-character string.
Web: Forward on No Reason Digit Pattern (Internal) EMS: Digit Pattern Forward No Reason **[DigitPatternForwardNoReason]**	Determines the digit pattern used by the PBX to indicate 'call forward with no reason' when the original call is received from an internal extension. The valid range is a 120-character string.
Web: Forward on Busy Digit Pattern (External) EMS: VM Digit Pattern On Busy External **[DigitPatternForwardOnBusyExt]**	Determines the digit pattern used by the PBX to indicate 'call forward on busy' when the original call is received from an external line (not an internal extension). The valid range is a 120-character string.
Web: Forward on No Answer Digit Pattern (External) EMS: VM Digit Pattern On No Answer Ext **[DigitPatternForwardOnNoAnswerExt]**	Determines the digit pattern used by the PBX to indicate 'call forward on no answer' when the original call is received from an external line (not an internal extension). The valid range is a 120-character string.
Web: Forward on Do Not Disturb Digit Pattern (External) EMS: VM Digit Pattern On DND External **[DigitPatternForwardOnDNDExt]**	Determines the digit pattern used by the PBX to indicate 'call forward on do not disturb' when the original call is received from an external line (not an internal extension). The valid range is a 120-character string.
Web: Forward on No Reason Digit Pattern (External) EMS: VM Digit Pattern No Reason External **[DigitPatternForwardNoReasonExt]**	Determines the digit pattern used by the PBX to indicate 'call forward with no reason' when the original call is received from an external line (not an internal extension). The valid range is a 120-character string.
Web: Internal Call Digit Pattern EMS: Digit Pattern Internal Call **[DigitPatternInternalCall]**	Determines the digit pattern used by the PBX to indicate an internal call. The valid range is a 120-character string.

Parameter	Description
Web: External Call Digit Pattern EMS: Digit Pattern External Call **[DigitPatternExternalCall]**	Determines the digit pattern used by the PBX to indicate an external call. The valid range is a 120-character string.
Web: Disconnect Call Digit Pattern EMS: Tel Disconnect Code **[TelDisconnectCode]**	Determines a digit pattern that when received from the Tel side, indicates the device to disconnect the call. The valid range is a 25-character string.
Web: Digit To Ignore Digit Pattern EMS: Digit To Ignore **[DigitPatternDigitToIgnore]**	A digit pattern that if received as Src (S) or Redirect (R) numbers is ignored and not added to that number. The valid range is a 25-character string.

6.7.5 Fax and Modem Parameters

The fax and modem parameters parameters are described in the table below.

Table 6-31: Fax and Modem Parameters

Parameter	Description
EMS: T38 Use RTP Port **[T38UseRTPPort]**	Defines the port (with relation to RTP port) for sending and receiving T.38 packets. ▪ **[0]** = Use the RTP port +2 to send/receive T.38 packets (default). ▪ **[1]** = Use the same port as the RTP port to send/receive T.38 packets. **Notes:** ▪ For this parameter to take effect, you must reset the device. ▪ When the device is configured to use V.152 to negotiate audio and T.38 coders, the UDP port published in SDP for RTP and for T38 must be different. Therefore, set the the parameter T38UseRTPPort to 0.
Web/EMS: T.38 Max Datagram Size **[T38MaxDatagramSize]**	Defines the maximum size of a T.38 datagram that the device can receive. This value is included in the outgoing SDP when T.38 is used. The valid range is 122 to 1,024. The default value is 122.
Web/EMS: T38 Fax Max Buffer **[T38FaxMaxBufferSize]**	Defines the maximum size (in bytes) of the device's T.38 buffer. This value is included in the outgoing SDP when T.38 is used for fax relay over IP. The valid range is 100 to 1,024. The default value is 1,024.
Web/EMS: Enable Fax Re-Routing **[EnableFaxReRouting]**	Enables or disables re-routing of Tel-to-IP calls that are identified as fax calls. ▪ **[0]** Disable = Disabled (default). ▪ **[1]** Enable = Enabled. If a CNG tone is detected on the Tel side of a Tel-to-IP call, the prefix "FAX" is appended to the destination number before routing and manipulations. A value of "FAX" entered as the destination number in the 'Outbound IP Routing Table' is then used to route the call and the destination number manipulation mechanism is used to remove the "FAX" prefix, if required.

Parameter	Description
	If the initial INVITE used to establish the voice call (not fax) was already sent, a CANCEL (if not connected yet) or a BYE (if already connected) is sent to tear down the voice call. **Notes:** ▪ To enable this feature, set the parameter CNGDetectorMode to 2 and the parameter IsFaxUsed to 1, 2, or 3. ▪ The "FAX" prefix in routing and manipulation tables is case-sensitive.
Web/EMS: Fax CNG Mode **[FaxCNGMode]**	Determines the device's behavior upon detection of a CNG tone. ▪ **[0]** = Does not send a SIP Re-INVITE upon detection of a fax CNG tone when the parameter CNGDetectorMode is set to 1 (default). ▪ **[1]** = Sends a SIP Re-INVITE upon detection of a fax CNG tone when the parameter CNGDetectorMode is set to 1.
Web: Detect Fax on Answer Tone EMS: Enables Detection of FAX on Answer Tone **[DetFaxOnAnswerTone]**	Determines when the device initiates a T.38 session for fax transmission. ▪ **[0]** Initiate T.38 on Preamble = The device to which the called fax is connected initiates a T.38 session on receiving HDLC Preamble signal from the fax (default). ▪ **[1]** Initiate T.38 on CED = The device to which the called fax is connected initiates a T.38 session on receiving a CED answer tone from the fax. This option can only be used to relay fax signals, as the device sends T.38 Re-INVITE on detection of any fax/modem Answer tone (2100 Hz, amplitude modulated 2100 Hz, or 2100 Hz with phase reversals). The modem signal fails when using T.38 for fax relay. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ This parameters is applicable only if the parameter IsFaxUsed is set to 1 (T.38 Relay) or 3 (Fax Fallback).

6.7.6 DTMF and Hook-Flash Parameters

The DTMF and hook-flash parameters parameters are described in the table below.

Table 6-32: DTMF and Hook-Flash Parameters

Parameter	Description
Hook-Flash Parameters	
Web/EMS: Hook-Flash Code **[HookFlashCode]**	For analog interfaces: Defines the digit pattern that when received from the Tel side, indicates a Hook Flash event. For digital interfaces: Determines the digit pattern used by the PBX to indicate a Hook Flash event. When this pattern is detected from the Tel side, the device responds as if a Hook Flash event occured and sends a SIP INFO message if the parameter HookFlashOption is set to 1, indicating Hook Flash. If configured and a Hook Flash indication is received from the IP side, the device generates this pattern to the Tel side. The valid range is a 25-character string. The default is a null string.
Web/EMS: Hook-Flash Option **[HookFlashOption]**	Determines the hook-flash transport type (i.e., method by which hook-flash is sent and received). ▪ **[0]** Not Supported = Hook-Flash indication isn't sent (default). ▪ **[1]** INFO = Send proprietary INFO message with Hook-Flash indication. ▪ **[4]** RFC 2833 ▪ **[5]** INFO (Lucent) = Send proprietary SIP INFO message with Hook-Flash indication. ▪ **[6]** INFO (NetCentrex) = Send proprietary SIP INFO message with Hook-Flash indication. The SIP INFO message is sent by the device as follows: Content-Type: application/dtmf-relay Signal=16 Where 16 is the DTMF code for hook flash ▪ **[7]** INFO (HUAWAEI) = Sends a SIP INFO message with Hook-Flash indication. The device sends the INFO message as follows: Content-Length: 17 Content-Type: application/sscc event=flashhook **Notes:** ▪ The RFC 2833 **[4]** option is currently not supported by digital interfaces. ▪ The device can interwork DTMF HookFlashCode to SIP INFO messages with Hook Flash indication (for digital interfaces). ▪ FXO interfaces support only the receipt of RFC 2833 Hook-Flash signals and INFO **[1]** type. ▪ FXS interfaces send Hook-Flash signals only if the parameter EnableHold is set to 0.

Parameter	Description
Web: Min. Flash-Hook Detection Period **[msec]** EMS: Min Flash Hook Time **[MinFlashHookTime]**	Defines the minimum time (in msec) for detection of a hook-flash event. Detection is guaranteed for hook-flash periods of at least 60 msec (when setting the minimum time to 25). Hook-flash signals that last a shorter period of time are ignored. The valid range is 25 to 300. The default value is 300. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ This parameter is applicable only to FXS interfaces. ▪ It's recommended to reduce the detection time by 50 msec from the desired value. For example, if you want to set the value to 200 msec, then enter 150 msec (i.e., 200 minus 50).
Web: Max. Flash-Hook Detection Period **[msec]** EMS: Flash Hook Period **[FlashHookPeriod]**	Defines the hook-flash period (in msec) for both Tel and IP sides (per device). For the IP side, it defines the hook-flash period that is reported to the IP. For the analog side, it defines the following: ▪ FXS interfaces: ✓ Maximum hook-flash detection period. A longer signal is considered an off-hook or on-hook event. ✓ Hook-flash generation period upon detection of a SIP INFO message containing a hook-flash signal. ▪ FXO interfaces: Hook-flash generation period. The valid range is 25 to 3,000. The default value is 700. **Notes:** ▪ For this parameter to take effect, you need to reset the device. ▪ For FXO interfaces, a constant of 100 msec must be added to the required hook-flash period. For example, to generate a 450 msec hook-flash, set this parameter to 550.
DTMF Parameters	
EMS: Use End of DTMF **[MGCPDTMFDetectionPoint]**	Defines when the detection of DTMF events is notified. ▪ **[0]** = DTMF event is reported at the end of a detected DTMF digit. ▪ **[1]** = DTMF event is reported at the start of a detected DTMF digit (default).
Web: Declare RFC 2833 in SDP EMS: Rx DTMF Option **[RxDTMFOption]**	Defines the supported Receive DTMF negotiation method. ▪ **[0]** No = Don't declare RFC 2833 telephony-event parameter in SDP. ▪ **[3]** Yes = Declare RFC 2833 telephony-event parameter in SDP (default). The device is designed to always be receptive to RFC 2833 DTMF relay packets. Therefore, it is always correct to include the 'telephony-event' parameter as default in the SDP. However, some devices use the absence of the 'telephony-event' in the SDP to decide to send DTMF digits in-band using G.711 coder. If this is the case, you can set this parameter to 0.

Parameter	Description
Web/EMS: Tx DTMF Option **[TxDTMFOption]**	Determines a single or several preferred transmit DTMF negotiation methods. ▪ **[0]** Not Supported = No negotiation - DTMF digits are sent according to the parameters DTMFTransportType and RFC2833PayloadType (default). ▪ **[1]** INFO (Nortel) = Sends DTMF digits according to IETF <draft-choudhuri-sip-info-digit-00>. ▪ **[2]** NOTIFY = Sends DTMF digits according to IETF <draft-mahy-sipping-signaled-digits-01>. ▪ **[3]** INFO (Cisco) = Sends DTMF digits according to Cisco format. ▪ **[4]** RFC 2833. ▪ **[5]** INFO (Korea) = Sends DTMF digits according to Korea Telecom format. **Notes:** ▪ DTMF negotiation methods are prioritized according to the order of their appearance. ▪ When out-of-band DTMF transfer is used (**[1], [2], [3],** or **[5]),** the parameter DTMFTransportType is automatically set to 0 (DTMF digits are erased from the RTP stream). ▪ When RFC 2833 (4) is selected, the device: **a.** Negotiates RFC 2833 payload type using local and remote SDPs. **b.** Sends DTMF packets using RFC 2833 payload type according to the payload type in the received SDP. **c.** Expects to receive RFC 2833 packets with the same payload type as configured by the parameter RFC2833PayloadType. **d.** Sends DTMF digits in transparent mode (as part of the voice stream). ▪ When TxDTMFOption is set to 0, the RFC 2833 payload type is set according to the parameter RFC2833PayloadType for both transmit and receive. ▪ The *ini* file table parameter TxDTMFOption can be repeated twice for configuring the DTMF transmit methods.
Web/EMS: Tx DTMF Option Table	
[TxDTMFOption]	This *ini* file table parameter configures up to two preferred transmit DTMF negotiation methods. The format of this parameter is as follows: [TxDTMFOption] FORMAT TxDTMFOption_Index = TxDTMFOption_Type; [\TxDTMFOption] For example: TxDTMFOption 0 = 1; TxDTMFOption 1 = 3; **Notes:** ▪ This parameter can include up two indices. ▪ For a description on using *ini* file table parameters, refer to "Configuring *ini* File Table Parameters" on page 222.

Parameter	Description
[DisableAutoDTMFMute]	Enables/disables the automatic muting of DTMF digits when out-of-band DTMF transmission is used. ▪ **[0]** = Automatic mute is used (default). ▪ **[1]** = No automatic mute of in-band DTMF. When this parameter is set to 1, the DTMF transport type is set according to the parameter DTMFTransportType and the DTMF digits aren't muted if out-of-band DTMF mode is selected (TxDTMFOption set to 1, 2 or 3). This enables the sending of DTMF digits in-band (transparent of RFC 2833) in addition to out-of-band DTMF messages. **Note:** Usually this mode is not recommended.
Web/EMS: Enable Digit Delivery to IP **[EnableDigitDelivery2IP]**	The Digit Delivery feature enables sending DTMF digits to the destination IP address after the Tel-to-IP call is answered. ▪ **[0]** Disable = Disabled (default). ▪ **[1]** Enable = Enable digit delivery to IP. To enable this feature, modify the called number to include at least one 'p' character. The device uses the digits before the 'p' character in the initial INVITE message. After the call is answered, the device waits for the required time (number of 'p' multiplied by 1.5 seconds), and then sends the rest of the DTMF digits using the method chosen (in-band or out-of-band). **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ The called number can include several 'p' characters (1.5 seconds pause), for example, 1001pp699, 8888p9p300.
Web: Enable Digit Delivery to Tel EMS: Enable Digit Delivery **[EnableDigitDelivery]**	Enables the Digit Delivery feature, which sends DTMF digits of the called number to the device's port (analog)/B-channel (digital) (phone line) after the call is answered (i.e., line is off-hooked for FXS, or seized for FXO) for IP-to-Tel calls. ▪ **[0]** Disable = Disabled (default). ▪ **[1]** Enable = Enable Digit Delivery feature for the FXO/FXS device. For digital interfaces: If the called number in IP-to-Tel call includes the characters 'w' or 'p', the device places a call with the first part of the called number (before 'w' or 'p') and plays DTMF digits after the call is answered. If the character 'w' is used, the device waits for detection of a dial tone before it starts playing DTMF digits. For example, if the called number is '1007766p100', the device places a call with 1007766 as the destination number, then after the call is answered it waits 1.5 seconds ('p') and plays the rest of the number (100) as DTMF digits. Additional examples: 1664wpp102, 66644ppp503, and 7774w100pp200. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ For analog interfaces: The called number can include characters 'p' (1.5 seconds pause) and 'd' (detection of dial tone). If character 'd' is used, it must be the first 'digit' in the called number. The character 'p' can be used several times.

Parameter	Description
	For example (for FXS/FXO interfaces), the called number can be as follows: d1005, dpp699, p9p300. To add the 'd' and 'p' digits, use the usual number manipulation rules. ▪ For analog interfaces: To use this feature with FXO interfaces, configure the device to operate in one-stage dialing mode. ▪ If this parameter is enabled, it is possible to configure the FXS/FXO interface to wait for dial tone per destination phone number (before or during dialing of destination phone number). Therefore, the parameter IsWaitForDialTone (configurable for the entire device) is ignored. ▪ For analog interfaces: The FXS interface send SIP 200 OK responses only after the DTMF dialing is complete.
Web/EMS: RFC 2833 Payload Type **[RFC2833PayloadType]**	The RFC 2833 DTMF relay dynamic payload type. The valid range is 96 to 99, and 106 to 127. The default is 96. The 100, 102 to 105 range is allocated for proprietary usage. **Notes:** ▪ Certain vendors (e.g., Cisco) use payload type 101 for RFC 2833. ▪ When RFC 2833 payload type negotiation is used (i.e., the parameter TxDTMFOption is set to 4), this payload type is used for the received DTMF packets. If negotiation isn't used, this payload type is used for receive and for transmit.
[ReplaceNumberSignWithEscapeChar]	Determines whether to replace the number sign (#) with the escape character (%23) in outgoing SIP messages for Tel-to-IP calls. ▪ **[0]** Disable (default) ▪ **[1]** Enable = All number signs #, received in the dialed DTMF digits are replaced in the outgoing SIP Request-URI and To headers with the escape sign %23. **Notes:** ▪ This parameter is applicable only if the parameter IsSpecialDigits is set 1. ▪ This parameter is applicable only to analog interfaces.
Web: Special Digit Representation EMS: Use Digit For Special DTMF **[UseDigitForSpecialDTMF]**	Defines the representation for 'special' digits ('*' and '#') that are used for out-of-band DTMF signaling (using SIP INFO/NOTIFY). ▪ **[0]** Special = Uses the strings '*' and '#' (default). ▪ **[1]** Numeric = Uses the numerical values 10 and 11.

6.7.7 Digit Collection and Dial Plan Parameters

The digit collection and dial plan parameters are described in the table below.

Table 6-33: Digit Collection and Dial Plan Parameters

Parameter	Description
Web/EMS: Dial Plan Index **[DialPlanIndex]**	Determines the Dial Plan index to use in the external Dial Plan file. The Dial Plan file is loaded to the device as a *.dat file (converted using the DConvert utility). The Dial Plan index can be defined globally or per Tel Profile. The valid value range is 0 to 7, where 0 denotes PLAN1, 1 denotes PLAN2, and so on. The default is -1, indicating that no Dial Plan file is used. **Notes:** ▪ If this parameter is configured to select a Dial Plan index, the settings of the parameter DigitMapping are ignored. ▪ This parameter is applicable also to ISDN with overlap dialing. ▪ For E1 CAS MFC-R2 variants (which don't support terminating digit for the called party number, usually I-15), this parameter and the DigitMapping parameter are ignored. Instead, you can define a Dial Plan template per trunk using the parameter CasTrunkDialPlanName_x (or in the 'Trunk Settings' page). ▪ For a detailed description of the Dial Plan file, refer to "External Dial Plan File" on page 488.
Web: Digit Mapping Rules EMS: Digit Map Patterns **[DigitMapping]**	Defines the digit map pattern (used to reduce the dialing period when ISDN overlap dialing is used in digital interfaces). If the digit string (i.e., dialed number) matches one of the patterns in the digit map, the device stops collecting digits and establishes a call with the collected number. The digit map pattern can contain up to 52 options (rules), each separated by a vertical bar (|). The maximum length of the entire digit pattern is 152 characters. The available notations include the following: ▪ **[n-m]:** Range of numbers (not letters). ▪ **.** (single dot): Repeat digits until next notation (e.g., T). ▪ **x:** Any single digit. ▪ **T:** Dial timeout (configured by the parameter TimeBetweenDigits). ▪ **S:** Immediately applies a specific rule that is part of a general rule. For example, if your digit map includes a general rule 'x.T' and a specific rule '11x', for the specific rule to take precedence over the general rule, append 'S' to the specific rule (i.e., '11xS'). An example of a digit map is shown below: 11xS|00T|**[1-**7]xxx|8xxxxxxx|#xxxxxxx|*xx|91xxxxxxxxxx|9011x.T In the example above, the last rule can apply to International numbers - 9 for dialing tone, 011 Country Code, and then any number of digits for the local number ('x.'). **Notes:** ▪ For ISDN interfaces, the digit map mechanism is applicable only when ISDN overlap dialing is used (ISDNRxOverlap is set to 1). ▪ If the parameter DialPlanIndex is configured (to select a Dial

Parameter	Description
	Plan index), then the parameter DigitMapping is ignored. ▪ For a detailed description of the digit mapping, refer to "Digit Mapping" on page 487.
Web: Max Digits in Phone Num EMS: Max Digits in Phone Number **[MaxDigits]**	Defines the maximum number of collected destination number digits that can be received (i.e., dialed) from the Tel side (analog) when Tel-to-IP ISDN overlap dialing is performed (digital). When the number of collected digits reaches this maximum, the device uses these digits for the called destination number. The valid range is 1 to 49. The default value is 5 for analog and 30 for digital. **Notes:** ▪ Digit Mapping Rules can be used instead. ▪ Dialing ends when any of the following scenarios occur: ✓ Maximum number of digits is dialed ✓ Interdigit Timeout (TimeBetweenDigits) expires ✓ Pound (#) key is pressed ✓ Digit map pattern is matched
Web: Inter Digit Timeout for Overlap Dialing **[sec]** EMS: Interdigit Timeout (Sec) **[TimeBetweenDigits]**	For analog interfaces: Defines the time (in seconds) that the device waits between digits that are dialed by the user. For ISDN overlap dialing: Defines the time (in seconds) that the device waits between digits that are received from the PSTN or IP during overlap dialing. When this inter-digit timeout expires, the device uses the collected digits to dial the called destination number. The valid range is 1 to 10. The default value is 4.

6.7.8 Coders and Profile Parameters

The profile parameters are described in the table below.

Table 6-34: Profile Parameters

Parameter	Description
Web: Coders Table/Coder Group Settings **EMS: Coders Group**	
[CodersGroup0] **[CodersGroup1]** **[CodersGroup2]** **[CodersGroup3]** **[CodersGroup4]**	This *ini* file table parameter defines the device's coders. Up to five groups of coders can be defined, where each group can consist of up to 10 coders. The first Coder Group is the default coder list and the default Coder Group. These Coder Groups can later be assigned to IP or Tel Profiles. The format of this parameter is as follows: [CodersGroup0] FORMAT CodersGroup0_Index = CodersGroup0_Name, CodersGroup0_pTime, CodersGroup0_rate, CodersGroup0_PayloadType, CodersGroup0_Sce; [\CodersGroup0] Where, ▪ Index = Coder entry 0-9, i.e., up to 10 coders per group.

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td></td><td>

Name = Coder name.
Ptime = Packetization time (ptime) - how many coder payloads are combined into a single RTP packet.
Rate = Packetization rate.
PayloadType = Identifies the format of the RTP payload.
Sce = Enables silence suppression:

✓ [0] Disabled (default)
✓ [1] Enabled

For example, below are defined two Coder Groups (0 and 1):
<pre>
[CodersGroup0]
FORMAT CodersGroup0_Index = CodersGroup0_Name,
CodersGroup0_pTime, CodersGroup0_rate,
CodersGroup0_PayloadType, CodersGroup0_Sce;
CodersGroup0 0 = g711Alaw64k, 20, 0, 255, 0;
CodersGroup0 1 = eg711Ulaw, 10, 0, 71, 0;
CodersGroup0 2 = eg711Ulaw, 10, 0, 71, 0;
[\CodersGroup0]
</pre>
<pre>
[CodersGroup1]
FORMAT CodersGroup1_Index = CodersGroup1_Name,
CodersGroup1_pTime, CodersGroup1_rate,
CodersGroup1_PayloadType, CodersGroup1_Sce;
CodersGroup1 0 = Transparent, 20, 0, 56, 0;
CodersGroup1 1 = g726, 20, 0, 23, 0;
[\CodersGroup1]
</pre>
The table below lists the supported coders:
<table>
<tr><th>Coder Name</th><th>Packetization Time (msec)</th><th>Rate (kbps)</th><th>Payload Type</th><th>Silence Suppression</th></tr>
<tr><td>G.711 A-law
[g711Alaw64k]</td><td>10, 20 (default), 30, 40, 50, 60, 80, 100, 120</td><td>Always 64</td><td>Always 8</td><td>Disable [0]
Enable [1]</td></tr>
<tr><td>G.711 U-law
[g711Ulaw64k]</td><td>10, 20 (default), 30, 40, 50, 60, 80, 100, 120</td><td>Always 64</td><td>Always 0</td><td>Disable [0]
Enable [1]</td></tr>
<tr><td>G.711A-law_VBD
[g711AlawVbd]</td><td>10, 20 (default), 30, 40, 50, 60, 80, 100, 120</td><td>Always 64</td><td>Dynamic (0-127)</td><td>N/A</td></tr>
<tr><td>G.711U-law_VBD
[g711UlawVbd]</td><td>10, 20 (default), 30, 40, 50, 60, 80, 100, 120</td><td>Always 64</td><td>Dynamic (0-127)</td><td>N/A</td></tr>
<tr><td>EG.711 A-law
[eg711Alaw]</td><td>10 (default), 20, 30</td><td>Always 64</td><td>Dynamic (96-127)</td><td>N/A</td></tr>
<tr><td>EG.711 U-law
[eg711Ulaw]</td><td>10 (default), 20, 30</td><td>Always 64</td><td>Dynamic (96-127)</td><td>N/A</td></tr>
<tr><td>G.722
[g722]</td><td>20 (default), 40, 60, 80, 100, 120</td><td>64 (default)</td><td>Always 9</td><td>N/A</td></tr>
<tr><td>G.723.1
[g7231]</td><td>30 (default), 60, 90, 120</td><td>5.3 [0], 6.3 [1] (default)</td><td>Always 4</td><td>Disable [0]
Enable [1]</td></tr>
</table>
</td></tr>
</table>

Parameter	Description

G.726 **[g726]**	10, 20 (default), 30, 40, 50, 60, 80, 100, 120	16 **[0]** (default), 24 **[1]**, 32 **[2]**, 40 **[3]**	Dynamic (0-127) Default is 23	Disable **[0]** Enable **[1]**
G.727 ADPCM	10, 20 (default), 30, 40, 50, 60, 80, 100, 120	16, 24, 32, 40	Dynamic (0-127)	Disable **[0]** Enable **[1]**
G.729 **[g729]**	10, 20 (default), 30, 40, 50, 60, 80, 100	Always 8	Always 18	Disable **[0]** Enable **[1]** Enable w/o Adaptations **[2]**
GSM-FR **[gsmFullRate]**	20 (default), 40, 60, 80	Always 13	Always 3	Disable **[0]** Enable **[1]**
GSM-EFR **[gsmEnhancedFullRate]**	0, 20 (default), 30, 40, 50, 60, 80, 100	12.2	Dynamic (0-127)	Disable **[0]** Enable **[1]**
MS-GSM **[gsmMS]**	40 (default)	Always 13	Always 3	Disable **[0]** Enable **[1]**
AMR **[Amr]**	20 (default)	4.75 **[0]**, 5.15 **[1]**, 5.90 **[2]**, 6.70 **[3]**, 7.40 **[4]**, 7.95 **[5]**, 10.2 **[6]**, 12.2 **[7]** (default)	Dynamic (0-127)	Disable **[0]** Enable **[1]**
QCELP **[QCELP]**	20 (default), 40, 60, 80, 100, 120	Always 13	Always 12	Disable **[0]** Enable **[1]**
EVRC **[Evrc]**	20 (default), 40,60, 80, 100	Variable **[0]** (default), 1/8 **[1]**, 1/2 **[3]**, Full **[4]**	Dynamic (0-127)	Disable **[0]** Enable **[1]**
iLBC **[iLBC]**	20 (default), 40, 60, 80, 100, 120	15 (default)	Dynamic (0-127)	Disable **[0]** Enable **[1]**
	30 (default), 60, 90, 120	13		
Transparent **[Transparent]**	20 (default), 40, 60, 80, 100, 120	Always 64	Dynamic (0-127)	Disable **[0]** Enable **[1]**
T.38 **[t38fax]**	N/A	N/A	N/A	N/A

Notes:

- The coder name is case-sensitive.
- Each coder type can appear only once per Coder Group.
- Only the packetization time of the first coder in the defined coder list is declared in INVITE/200 OK SDP, even if multiple coders are defined.
- The device always uses the packetization time requested by the remote side for sending RTP packets. If not specified, the

Parameter	Description
	packetization time is assigned the default value. ▪ The value of several fields is hard-coded according to common standards (e.g., payload type of G.711 U-law is always 0). Other values can be set dynamically. If no value is specified for a dynamic field, a default value is assigned. If a value is specified for a hard-coded field, the value is ignored. ▪ If silence suppression is not defined for a specific coder, the value defined by the parameter EnableSilenceCompression is used. ▪ If G.729 is selected and silence suppression is enabled (for this coder), the device includes the string 'annexb=no' in the SDP of the relevant SIP messages. If silence suppression is set to 'Enable w/o Adaptations', 'annexb=yes' is included. An exception is when the remote device is a Cisco gateway (IsCiscoSCEMode). ▪ The coder G.722 provides Packet Loss Concealment (PLC) capabilities, ensuring higher voice quality. ▪ Both GSM-FR and MS-GSM coders use Payload Type 3. When using SDP, it isn't possible to differentiate between the two. Therefore, it is recommended not to select both coders simultaneously. ▪ For an explanation on V.152 support (and implementation of T.38 and VBD coders), refer to "V.152 Support" on page 563. ▪ For a description of using *ini* file table parameters, refer to "Configuring *ini* File Table Parameters" on page 222.
Web: IP Profile Settings Table **EMS: Protocol Definition > IP Profile**	
[IPProfile]	This *ini* file table parameter configures the IP Profile table. Each IP Profile ID includes a set of parameters (which are typically configured separately using their individual "global" parameters). You can later assign these IP Profiles to outbound IP routing rules (Prefix parameter), inbound IP routing rules (PSTNPrefix parameter), and IP Groups (IPGroup parameter). The format of this parameter is as follows: [IPProfile] FORMAT IPProfile_Index = IPProfile_ProfileName, IPProfile_IpPreference, IPProfile_CodersGroupID, IPProfile_IsFaxUsed, IPProfile_JitterBufMinDelay, IPProfile_JitterBufOptFactor, IPProfile_IPDiffServ, IPProfile_SigIPDiffServ, IpProfile_SCE, IPProfile_RTPRedundancyDepth, IPProfile_RemoteBaseUDPPort, IPProfile_CNGmode, IPProfile_VxxTransportType, IPProfile_NSEMode, IpProfile_IsDTMFUsed, IPProfile_PlayRBTone2IP, IPProfile_EnableEarlyMedia, IPProfile_ProgressIndicator2IP, IPProfile_EnableEchoCanceller, IPProfile_CopyDest2RedirectNumber, IPProfile_MediaSecurityBehaviour, IPProfile_CallLimit, IPProfile_ DisconnectOnBrokenConnection, IPProfile_FirstTxDtmfOption, IPProfile_SecondTxDtmfOption, IPProfile_RxDTMFOption, IpProfile_EnableHold, IpProfile_InputGain, IpProfile_VoiceVolume, IpProfile_AddIEInSetup, IpProfile_SBCExtensionCodersGroupID, IPProfile_MediaIPVersionPreference, IPProfile_TranscodingMode, IpProfile_SBCAllowedCodersGroupID, IpProfile_SBCAllowedCodersMode, IpProfile_SBCMediaSecurityBehaviour, IpProfile_SBCRFC2833Behavior, IpProfile_SBCAlternativeDTMFMethod, IpProfile_SBCAssertIdentity; [\IPProfile]

Parameter	Description
	For example: IPProfile 0 = Sevilia, 1, 1, 0, 10, 10, 46, 40, 0, 0, 0, 0, 2, 0, 0, 0, 0, -1, 1, 0, 0, -1, 1, -1, -1, 1, 1, 0, 0, , -1, 4294967295, 0; **Notes:** ▪ You can configure up to nine IP Profiles (i.e., indices 1 through 9). ▪ The following parameters are not applicable: SBCExtensionCodersGroupID, TranscodingMode SBCAllowedCodersGroupID, SBCAllowedCodersMode, SBCMediaSecurityBehaviour, SBCRFC2833Behavior, SBCAlternativeDTMFMethod, and SBCAssertIdentity. ▪ The parameter MediaIPVersionPreference is not applicable. ▪ The parameter IsDTMFUsed is not applicable (deprecated). ▪ The parameter IpPreference determines the priority of the IP Profile (1 to 20, where 20 is the highest preference). If both IP and Tel Profiles apply to the same call, the coders and common parameters (i.e., parameters configurable in both IP and Tel Profiles) of the preferred profile are applied to that call. If the Tel and IP Profiles are identical, the Tel Profile parameters take precedence. ▪ To use the settings of the corresponding global parameter, enter the value -1. ▪ The parameter CallLimit defines the maximum number of concurrent calls allowed for that Profile. If the Profile is set to some limit, the device maintains the number of concurrent calls (incoming and outgoing) pertaining to the specific Profile. A limit value of **[-1**] indicates that there is no limitation on calls (default). A limit value of **[0]** indicates that all calls are rejected. When the number of concurrent calls is equal to the limit, the device rejects any new incoming and outgoing calls pertaining to that profile. ▪ RxDTMFOption configures the received DTMF negotiation method: **[-1**] not configured, use the global parameter; **[0]** don't declare RFC 2833; **[1]** declare RFC 2833 payload type is SDP. ▪ FirstTxDtmfOption and SecondTxDtmfOption configures the transmit DTMF negotiation method: **[-1**] not configured, use the global parameter; for the remaining options, refer to the global parameter. ▪ IP Profiles can also be used when operating with a Proxy server (set the parameter AlwaysUseRouteTable to 1). ▪ For a detailed description of each parameter, refer to its corresponding global parameter. ▪ For a description of using *ini* file table parameters, refer to "Configuring *ini* File Table Parameters" on page 222.
Web: Tel Profile Settings Table **EMS: Protocol Definition > Telephony Profile**	
[TelProfile]	This *ini* file table parameter configures the Tel Profile table. Each Tel Profile ID includes a set of parameters (which are typically configured separately using their individual, "global" parameters). You can later assign these Tel Profile IDs to other elements such as in the Trunk Group Table (TrunkGroup parameter). Therefore, Tel Profiles allow you to apply the same settings of a group of parameters to multiple channels, or apply specific settings to different channels. The format of this parameter is as follows:

Parameter	Description
	[TelProfile] FORMAT TelProfile_Index = TelProfile_ProfileName, TelProfile_TelPreference, TelProfile_CodersGroupID, TelProfile_IsFaxUsed, TelProfile_JitterBufMinDelay, TelProfile_JitterBufOptFactor, TelProfile_IPDiffServ, TelProfile_SigIPDiffServ, TelProfile_DtmfVolume, TelProfile_InputGain, TelProfile_VoiceVolume, TelProfile_EnableReversePolarity, TelProfile_EnableCurrentDisconnect, TelProfile_EnableDigitDelivery, TelProfile_EnableEC, TelProfile_MWIAnalog, TelProfile_MWIDisplay, TelProfile_FlashHookPeriod, TelProfile_EnableEarlyMedia, TelProfile_ProgressIndicator2IP, TelProfile_TimeForReorderTone, TelProfile_EnableDIDWink, TelProfile_IsTwoStageDial, TelProfile_DisconnectOnBusyTone, TelProfile_EnableVoiceMailDelay, TelProfile_DialPlanIndex, TelProfile_Enable911PSAP, TelProfile_SwapTelToIpPhoneNumbers, TelProfile_EnableAGC, TelProfile_ECNlpMode; [\TelProfile] For example: TelProfile 1 = ITSP_audio, 1, 0, 0, 10, 10, 46, 40, -11, 0, 0, 0, 0, 0, 1, 0, 0, 700, 0, -1, 255, 0, 1, 1, 1, -1, 1, 0, 0, 0; **Notes:** ▪ You can configure up to nine Tel Profiles (i.e., indices 1 through 9). ▪ The following parameters are applicable only to analog interfaces: EnableReversePolarity, EnableCurrentDisconnect, MWIAnalog, MWIDisplay, EnableDIDWink, IsTwoStageDial, DisconnectOnBusyTone, and Enable911PSAP. ▪ The parameter IpPreference determines the priority of the Tel Profile (1 to 20, where 20 is the highest preference). If both IP and Tel Profiles apply to the same call, the coders and common parameters (i.e., parameters configurable in both IP and Tel Profiles) of the preferred profile are applied to that call. If the Tel and IP Profiles are identical, the Tel Profile parameters take precedence. ▪ The parameter EnableVoiceMailDelay is applicable only if voice mail is enabled globally (using the parameter VoiceMailInterface). ▪ To use the settings of the corresponding global parameter, enter the value -1. ▪ For a detailed description of each parameter, refer to its corresponding "global" parameter. ▪ For a description of using *ini* file table parameters, refer to "Configuring *ini* File Table Parameters" on page 222.

6.8 Supplementary Services Parameters

This subsection describes the device's supplementary telephony services parameters.

6.8.1 Caller ID Parameters

The caller ID parameters are described in the table below.

Table 6-35: Caller ID Parameters

Parameter	Description
Web: Caller ID Permissions Table **EMS: SIP Endpoints > Caller ID**	
[EnableCallerID]	This *ini* file table parameter configures Caller ID permissions. It allows you to enable or disable (per port) Caller ID generation (for FXS interfaces) and detection (for FXO interfaces). The format of this parameter is as follows: [EnableCallerID] FORMAT EnableCallerID_Index = EnableCallerID_IsEnabled, EnableCallerID_Module, EnableCallerID_Port; [\EnableCallerID] Where, ▪ IsEnabled: ✓ **[0]** Disable = disables Caller ID (default). ✓ **[1]** Enable = enables Caller ID generation (FXS) or detection (FXO). ▪ Module = Module number (where 1 depicts the module in Slot 1). ▪ Port = Port number (where 1 depicts Port 1 of a module). For example: EnableCallerID 0 = 1,3,1; (caller ID enabled on Port 1 of Module 3) EnableCallerID 1 = 0,3,2; (caller ID disabled on Port 2 of Module 3) **Notes:** ▪ The indexing of this parameter starts at 0. ▪ If a port is not configured, its Caller ID generation/detection is determined according to the global parameter EnableCallerID. ▪ For configuring this table using the Web interface, refer to Configuring Caller ID Permissions on page 176. ▪ For an explanation on using *ini* file table parameters, refer to Configuring *ini* File Table Parameters on page 222.
Web: Caller Display Information Table **EMS: SIP Endpoints > Caller ID**	
[CallerDisplayInfo]	This *ini* file table parameter enables the device to send Caller ID information to IP when a call is made. The called party can use this information for caller identification. The information configured in this table is sent in the SIP INVITE message's

Parameter	Description
	From header. The format of this parameter is as follows: [CallerDisplayInfo] FORMAT CallerDisplayInfo_Index = CallerDisplayInfo_DisplayString, CallerDisplayInfo_IsCidRestricted, CallerDisplayInfo_Module, CallerDisplayInfo_Port; [\CallerDisplayInfo] Where, ▪ DisplayString = Caller ID string (up to 18 characters). ▪ IsCidRestricted = ✓ **[0]** Allowed = sends the defined caller ID string when a Tel-to-IP call is made using the corresponding device port (default). ✓ **[1]** Restricted = does not send the defined caller ID string. ▪ Module = Module number (where 1 depicts the module in Slot 1). ▪ Port = Port number (where 1 depicts Port 1 of a module). For example: CallerDisplayInfo 0 = Susan C.,0,1,1; ("Susan C." is sent as the Caller ID for Port 1 of Module 1) CallerDisplayInfo 1 = Mark M.,0,1,2; ("Mark M." is sent as Caller ID for Port 2 of Module 1) **Notes:** ▪ The indexing of this *ini* file table parameter starts at 0. ▪ When FXS ports receive 'Private' or 'Anonymous' strings in the SIP From header, the calling name or number is not sent to the Caller ID display. ▪ If the Caller ID name is detected on an FXO line (the parameter EnableCallerID is set to 1), it is used instead of the Caller ID name defined in this table parameter. ▪ When the parameter CallerDisplayInfo_IsCidRestricted is set to 1 (Restricted), the Caller ID is sent to the remote side using only the SIP headers P-Asserted-Identity and P-Preferred-Identity (AssertedIdMode). ▪ The value of the parameter CallerDisplayInfo_IsCidRestricted is overridden by the parameter SourceNumberMapIp2Tel_IsPresentationRestricted in the 'Source Number Manipulation' table (table parameter SourceNumberMapIP2Tel). ▪ For configuring this table using the Web interface, refer to Configuring Caller Display Information on page 173. ▪ For an explanation on using *ini* file table parameters, refer to Configuring *ini* File Table Parameters on page 222.

Parameter	Description
Web/EMS: Enable Caller ID **[EnableCallerID]**	Determines whether Caller ID is enabled. ▪ **[0]** Disable = Disable the Caller ID service (default). ▪ **[1]** Enable = Enable the Caller ID service. If the Caller ID service is enabled, then for FXS interfaces, calling number and Display text (from IP) are sent to the device's port. For FXO interfaces, the Caller ID signal is detected and sent to IP in the SIP INVITE message (as 'Display' element). For information on the Caller ID table, refer to Configuring Caller Display Information on page 173. To disable/enable caller ID generation per port, refer to Configuring Call Forward on page 175.
Web: Caller ID Type EMS: Caller id Types **[CallerIDType]**	Defines one of the following standards for detection (FXO) and generation (FXS) of Caller ID, and detection (FXO) generation (FXS) of MWI (when specified) signals: ▪ **[0]** Standard Bellcore = Caller ID and MWI (default) ▪ **[1]** Standard ETSI = Caller ID and MWI ▪ **[2]** Standard NTT ▪ **[4]** Standard BT = Britain ▪ **[16]** Standard DTMF Based ETSI ▪ **[17]** Standard Denmark = Caller ID and MWI ▪ **[18]** Standard India ▪ **[19]** Standard Brazil **Notes:** ▪ Typically, the Caller ID signals are generated/detected between the first and second rings. However, sometimes the Caller ID is detected before the first ring signal (in such a scenario, configure the parameter RingsBeforeCallerID to 0). ▪ Caller ID detection for Britain **[4]** is not supported on the device's FXO ports. Only FXS ports can generate the Britain **[4]** Caller ID. ▪ To select the Bellcore Caller ID sub standard, use the parameter BellcoreCallerIDTypeOneSubStandard. To select the ETSI Caller ID substandard, use the parameter ETSICallerIDTypeOneSubStandard. ▪ To select the Bellcore MWI sub standard, use the parameter BellcoreVMWITypeOneStandard. To select the ETSI MWI sub standard, use the parameter ETSIVMWITypeOneStandard. ▪ If you define Caller ID Type as NTT **[2],** you need to define the NTT DID signaling form (FSK or DTMF) using the parameter NTTDIDSignallingForm.

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>Web: Enable FXS Caller ID Category Digit For Brazil Telecom
[AddCPCPrefix2BrazilCallerID]</td><td>Enables the interworking of Calling Party Category (cpc) code from SIP INVITE messages to FXS Caller ID first digit.
▪ [0] Disable (default)
▪ [1] Enable = Interworking of CPC is performed
When this parameter is enabled, the device sends the Caller ID number (calling number) with the cpc code (received in the SIP INVITE message) to the device's FXS port. The cpc code is added as a prefix to the caller ID (after IP-to-Tel calling number manipulation). For example, assuming that the incoming INVITE contains the following From (or P-Asserted-Id) header:
From:<sip:+551137077801;cpc=payphone@10.20.7.35>;tag=53700
The calling number manipulation removes "+55" (leaving 10 digits), and then adds the prefix 7, the cpc code for payphone user. Therefore, the Caller ID number that is sent to the FXS port, in this example is 71137077801.
If the incoming INVITE message doesn't contain the 'cpc' parameter, nothing is added to the Caller ID number.
<table>
<tr><th>CPC Value in Received INVITE</th><th>CPC Code Prefixed to Caller ID (Sent to FXS Endpoint)</th><th>Description</th></tr>
<tr><td>cpc=unknown</td><td>1</td><td>Unknown user</td></tr>
<tr><td>cpc=subscribe</td><td>1</td><td>-</td></tr>
<tr><td>cpc=ordinary</td><td>1</td><td>Ordinary user</td></tr>
<tr><td>cpc=priority</td><td>2</td><td>Pre-paid user</td></tr>
<tr><td>cpc=test</td><td>3</td><td>Test user</td></tr>
<tr><td>cpc=operator</td><td>5</td><td>Operator</td></tr>
<tr><td>cpc=data</td><td>6</td><td>Data call</td></tr>
<tr><td>cpc=payphone</td><td>7</td><td>Payphone user</td></tr>
</table>
Notes:
▪ This parameter is applicable only to FXS interfaces.
▪ For this parameter to be enabled, you must also set the parameter EnableCallingPartyCategory to 1.</td></tr>
<tr><td>[EnableCallerIDTypeTwo]</td><td>Disables the generation of Caller ID type 2 when the phone is off-hooked. Caller ID type 2 (also known as off-hook Caller ID) is sent to a currently busy telephone to display the caller ID of the waiting call.
▪ [0] = Caller ID type 2 isn't played.
▪ [1] = Caller ID type 2 is played (default).</td></tr>
</table>

Parameter	Description
EMS: Caller ID Timing Mode **[AnalogCallerIDTimingMode]**	Determines when Caller ID is generated. ▪ **[0]** = Caller ID is generated between the first two rings (default). ▪ **[1]** = The device attempts to find an optimized timing to generate the Caller ID according to the selected Caller ID type. **Notes:** ▪ This parameter is applicable only to FXS interfaces. ▪ If this parameter is set to 1 and used with distinctive ringing, the Caller ID signal doesn't change the distinctive ringing timing. ▪ For this parameter to take effect, a device reset is required.
EMS: Bellcore Caller ID Type One Sub Standard **[BellcoreCallerIDTypeOneSubStandard]**	Selects the Bellcore Caller ID sub-standard. ▪ **[0]** = Between rings (default). ▪ **[1]** = Not ring related. **Note:** For this parameter to take effect, a device reset is required.
EMS: ETSI Caller ID Type One Sub Standard **[ETSICallerIDTypeOneSubStandard]**	Selects the ETSI FSK Caller ID Type 1 sub-standard (FXS only). ▪ **[0]** = ETSI between rings (default). ▪ **[1]** = ETSI before ring DT_AS. ▪ **[2]** = ETSI before ring RP_AS. ▪ **[3]** = ETSI before ring LR_DT_AS. ▪ **[4]** = ETSI not ring related DT_AS. ▪ **[5]** = ETSI not ring related RP_AS. ▪ **[6]** = ETSI not ring related LR_DT_AS. **Note:** For this parameter to take effect, a device reset is required.
Web: Asserted Identity Mode EMS: Asserted ID Mode **[AssertedIdMode]**	Determines whether P-Asserted-Identity or P-Preferred-Identity is used in the generated INVITE request for Caller ID (or privacy). ▪ **[0]** Disabled = None (default) ▪ **[1]** Adding PAsserted Identity ▪ **[2]** Adding PPreferred Identity This parameter determines the header (P-Asserted-Identity or P-Preferred-Identity) used in the generated INVITE request. The header also depends on the calling Privacy (allowed or restricted). These headers are used to present the originating party's Caller ID. The Caller ID is composed of a Calling Number and (optionally), a Calling Name. These headers are used together with the Privacy header. If Caller ID is restricted (i.e., P-Asserted-Identity is not sent), the Privacy header includes the value 'id' ('Privacy: id'). Otherwise, for allowed Caller ID, 'Privacy: none' is used. If Caller ID is

Parameter	Description
	restricted (received from Tel or configured in the device), the From header is set to <anonymous@anonymous.invalid>. The 200 OK response can contain the connected party CallerID - Connected Number and Connected Name. For example, if the call is answered by the device, the 200 OK response includes the P-Asserted-Identity with Caller ID. The device interworks (in some ISDN variants), the Connected Party number and name from Q.931 Connect message to SIP 200 OK with the P-Asserted-Identity header. In the opposite direction, if the ISDN device receives a 200 OK with P-Asserted-Identity header, it interworks it to the Connected party number and name in the Q.931 Connect message, including its privacy.
Web: Caller ID Transport Type EMS: Transport Type **[CallerIDTransportType]**	Determines the device's behavior for Caller ID detection. ▪ **[0]** Disable = The caller ID signal is not detected - DTMF digits remain in the voice stream. ▪ **[1]** Relay = (Currently not applicable.) ▪ **[3]** Mute = The caller ID signal is detected from the Tel/PSTN side and then erased from the voice stream (default). **Note:** Caller ID detection is applicable only to FXO interfaces.

6.8.2 Call Waiting Parameters

The call waiting parameters are described in the table below.

Table 6-36: Call Waiting Parameters

Parameter	Description
Web/EMS: Enable Call Waiting **[EnableCallWaiting]**	Determines whether Call Waiting is enabled. ▪ **[0]** Disable = Disable the Call Waiting service. ▪ **[1]** Enable = Enable the Call Waiting service (default). If enabled, when an FXS interface receives a call on a busy endpoint, it responds with a 182 response (and not with a 486 busy). The device plays a call waiting indication signal. When hook-flash is detected, the device switches to the waiting call. The device that initiated the waiting call plays a Call Waiting Ringback tone to the calling party after a 182 response is received. **Notes:** ▪ The device's Call Progress Tones (CPT) file must include a Call Waiting Ringback tone (caller side) and a Call Waiting tone (called side, FXS only). ▪ The EnableHold parameter must be enabled on both the calling and the called side. ▪ For analog interfaces: You can use the *ini* file table parameter CallWaitingPerPort to enable Call Waiting per port.

Parameter	Description
	▪ For information on the Call Waiting feature, refer to "Call Waiting" on page 571. ▪ For information on the Call Progress Tones file, refer to Configuring the Call Progress Tones File.
EMS: Send 180 For Call Waiting **[Send180ForCallWaiting]**	Determines the SIP response code for indicating Call Waiting. ▪ **[0]** = Use 182 Queued response to indicate call waiting (default). ▪ **[1]** = Use 180 Ringing response to indicate call waiting.
Web: Call Waiting Table **EMS: SIP Endpoints > Call Waiting**	
[CallWaitingPerPort]	This *ini* file table parameter configures call waiting per FXS port. The format of this parameter is as follows: [CallWaitingPerPort] FORMAT CallWaitingPerPort_Index = CallWaitingPerPort_IsEnabled, CallWaitingPerPort_Module, CallWaitingPerPort_Port; [\CallWaitingPerPort] Where, ▪ IsEnabled: ✓ **[0]** Disable = no call waiting for the specific port. ✓ **[1]** Enable = enables call waiting for the specific port. When the FXS device receives a call on a busy endpoint (port), it responds with a SIP 182 response (and not with a 486 busy). The device plays a call waiting indication signal. When hook-flash is detected, the device switches to the waiting call. The device that initiates the waiting call plays a Call Waiting Ringback tone to the calling party after a 182 response is received. ▪ Port = Port number. ▪ Module = Module number. For example: CallWaitingPerPort 0 = 0,1,1; (call waiting disabled for Port 1 of Module 1) CallWaitingPerPort 1 = 1,1,2; (call waiting enabled for Port 2 of Module 1) **Notes:** ▪ This parameter is applicable only to FXS ports. ▪ If this parameter is not configured (default), call waiting is determined according to the global parameter EnableCallWaiting. ▪ The device's CPT file must include a 'call waiting Ringback' tone (caller side) and a 'call waiting' tone (called side, FXS interfaces only). ▪ The EnableHold parameter must be enabled on both the calling and the called sides. ▪ For configuring this table using the Web interface, refer to Configuring Call Waiting on page 177.

Parameter	Description
	▪ For a description on using *ini* file table parameters, refer to Configuring *ini* File Table Parameters on page 222.
Web: Number of Call Waiting Indications EMS: Call Waiting Number of Indications **[NumberOfWaitingIndications]**	Number of Call Waiting indications that are played to the called telephone that is connected to the device for Call Waiting. The valid range is 1 to 100 indications. The default value is 2. **Note:** This parameter is applicable only to FXS ports.
Web: Time Between Call Waiting Indications EMS: Call Waiting Time Between Indications **[TimeBetweenWaitingIndications]**	Time (in seconds) between consecutive call waiting indications for call waiting. The valid range is 1 to 100. The default value is 10. **Note:** This parameter is applicable only to FXS ports.
Web/EMS: Time Before Waiting Indications **[TimeBeforeWaitingIndication]**	Defines the interval (in seconds) before a call waiting indication is played to the port that is currently in a call. The valid range is 0 to 100. The default time is 0 seconds. **Note:** This parameter is applicable only to FXS ports.
Web/EMS: Waiting Beep Duration **[WaitingBeepDuration]**	Duration (in msec) of call waiting indications that are played to the port that is receiving the call. The valid range is 100 to 65535. The default value is 300. **Note:** This parameter is applicable only to FXS ports.
EMS: First Call Waiting Tone ID **[FirstCallWaitingToneID]**	Determines the index of the first Call Waiting Tone in the CPT file. This feature enables the called party to distinguish between different call origins (e.g., external versus internal calls). There are three ways to use the distinctive call waiting tones: ▪ Playing the call waiting tone according to the SIP Alert-Info header in the received 180 Ringing SIP response. The value of the Alert-Info header is added to the value of the FirstCallWaitingToneID parameter. ▪ Playing the call waiting tone according to PriorityIndex in the ToneIndex *ini* file table parameter. ▪ Playing the call waiting tone according to the parameter “CallWaitingTone#' of a SIP INFO message. The device plays the tone received in the 'play tone CallWaitingTone#' parameter of an INFO message plus the value of this parameter minus 1. The valid range is -1 to 1,000. The default value is -1 (i.e., not used). **Notes:** ▪ This parameter is applicable only to analog interfaces. ▪ It is assumed that all Call Waiting Tones are defined in sequence in the CPT file. ▪ SIP Alert-Info header examples: ✓ Alert-Info:<Bellcore-dr2> ✓ Alert-Info:<http://…/Bellcore-dr2> (where "dr2" defines call waiting tone #2) ▪ The SIP INFO message is according to Broadsoft's application server definition. Below is an example of such an INFO message:

Parameter	Description
	INFO sip:06@192.168.13.2:5060 SIP/2.0 Via:SIP/2.0/UDP 192.168.13.40:5060;branch=z9hG4bK040066422630 From: <sip:4505656002@192.168.13.40:5060>;tag=1455352915 To: <sip:06@192.168.13.2:5060> Call-ID:0010-0008@192.168.13.2 CSeq:342168303 INFO Content-Length:28 Content-Type:application/broadsoft play tone CallWaitingTone1

6.8.3 Call Forwarding Parameters

The call forwarding parameters are described in the table below.

Table 6-37: Call Forwarding Parameters

Parameter	Description
Web: Enable Call Forward **[EnableForward]**	Determines whether Call Forward is enabled. ▪ **[0]** Disable = Disable the Call Forward service. ▪ **[1]** Enable = Enable Call Forward service (using REFER) (default). For FXS interfaces, the 'Call Forward' table (FwdInfo parameter) must be defined to use the Call Forward service. **Note:** To use this service, the devices at both ends must support this option.
Web: Call Forwarding Table **EMS: SIP Endpoints > Call Forward**	
[FwdInfo]	This *ini* file table parameter forwards (redirects) IP-to-Tel calls (using SIP 302 response) to other device ports or an IP destination, based on the device's port to which the call was originally routed. The format of this parameter is as follows: [FwdInfo] FORMAT FwdInfo_Index = FwdInfo_Type, FwdInfo_Destination, FwdInfo_NoReplyTime, FwdInfo_Module, FwdInfo_Port; [\FwdInfo] Where, ▪ Type = the scenario for forwarding the call: ✓ **[0]** Deactivate = Don't forward incoming calls (default). ✓ **[1]** On Busy = Forward incoming calls when the port is busy. ✓ **[2]** Unconditional = Always forward incoming calls. ✓ **[3]** No Answer = Forward incoming calls that are not answered within the time specified in the 'Time for No Reply Forward' field. ✓ **[4]** On Busy or No Answer = Forward incoming calls when the port is busy or when calls are not answered within the time specified in the 'Time for No Reply Forward' field. ✓ **[5]** Do Not Disturb = Immediately reject incoming calls.

Parameter	Description
	▪ Destination = Telephone number or URI (<number>@<IP address>) to where the call is forwarded. ▪ NoReplyTime = Timeout (in seconds) for No Reply. If you have set the Forward Type for this port to No Answer **[3],** enter the number of seconds the device waits before forwarding the call to the specified phone number. ▪ Module = Module number (where 1 depicts the module in Slot 1). ▪ Port = Port number (where 1 depicts Port 1 of a module). For example: ▪ Below configuration forwards calls originally destined to Port 1 of Module 1 to "1001" upon On Busy: FwdInfo 0 = 1,1001,30,1,1; ▪ Below configuration forwards calls originally destined to Port 2 of Module 1 to an IP address upon On Busy: FwdInfo 1 = 1,2003@10.5.1.1,30,1,2; **Notes:** ▪ The indexing of this parameter starts at 0. ▪ Ensure that the Call Forward feature is enabled (default) for the settings of this table parameter to take effect. To enable Call Forwarding, use the parameter EnableForward. ▪ If the parameter FwdInfo_Destination only contains a telephone number and a Proxy isn't used, the 'forward to' phone number must be specified in the 'Outbound IP Routing Table' (Prefix *ini* file parameter). ▪ For configuring this table using the Web interface, refer to Configuring Call Forward on page 175. ▪ For an explanation on using *ini* file table parameters, refer to Configuring *ini* File Table Parameters on page 222.
Call Forward Reminder Ring Parameters **Notes:** ▪ These parameters are applicable only to FXS interfaces. ▪ For a description of this feature, refer to Call Forward Reminder Ring on page 569.	
Web: Enable NRT Subscription **[EnableNRTSubscription]**	Enables Endpoint subscription for Ring reminder event notification feature. ▪ **[0]** Disable (default) ▪ **[1]** Enable
Web: AS Subscribe IPGroupID **[ASSubscribeIPGroupID]**	Defines the IP Group ID that contains the Application server for Subscription. The valid value range is 1 to 8. The default is -1 (i.e., not configured).
Web: NRT Retry Subscription Time **[NRTRetrySubscriptionTime]**	Defines the Retry period (in seconds) for Dialog subscription if a previous request failed. The valid value range is 10 to 7200. The default is 120.
Web: Call Forward Ring Tone ID **[CallForwardRingToneID]**	Defines the ringing tone type played when call forward notification is accepted. The valid value range is 1 to 5. The default is 1.

6.8.4 Message Waiting Indication Parameters

The message waiting indication (MWI) parameters are described in the table below.

Table 6-38: MWI Parameters

Parameter	Description
Web: Enable MWI EMS: MWI Enable **[EnableMWI]**	Enables Message Waiting Indication (MWI). ▪ **[0]** Disable = Disabled (default). ▪ **[1]** Enable = MWI service is enabled. **Notes:** ▪ This parameter is applicable only to FXS interfaces. ▪ The device supports only the receipt of SIP MWI NOTIFY messages (the device doesn't generate these messages). ▪ For detailed information on MWI, refer to "Message Waiting Indication" on page 572.
Web/EMS: MWI Analog Lamp **[MWIAnalogLamp]**	Enables the visual display of MWI. ▪ **[0]** Disable = Disable (default). ▪ **[1]** Enable = Enables visual MWI by supplying line voltage of approximately 100 VDC to activate the phone's lamp. **Note:** This parameter is applicable only for FXS interfaces.
Web/EMS: MWI Display **[MWIDisplay]**	Determines whether MWI information is sent to the phone display. ▪ **[0]** Disable = MWI information isn't sent to display (default). ▪ **[1]** Enable = The device generates an MWI message (determined by the parameter CallerIDType), which is displayed on the MWI display. **Note:** This parameter is applicable only to FXS interfaces.
Web: Subscribe to MWI EMS: Enable MWI Subscription **[EnableMWISubscription]**	Enables subscription to an MWI server. ▪ **[0]** No = Disables MWI subscription (default). ▪ **[1]** Yes = Enables subscription to an MWI server (defined by the parameter MWIServerIP address). **Note:** To configure whether the device subscribes per endpoint or per the entire device, use the parameter SubscriptionMode.
Web: MWI Server IP Address EMS: MWI Server IP **[MWIServerIP]**	MWI server's IP address. If provided, the device subscribes to this IP address. The MWI server address can be configured as a numerical IP address or as a domain name. If not configured, the Proxy IP address is used instead.

Parameter	Description
Web/EMS: MWI Server Transport Type **[MWIServerTransportType]**	Determines the transport layer used for outgoing SIP dialogs initiated by the device to the MWI server. ▪ **[-1]** Not Configured (default) ▪ **[0]** UDP ▪ **[1]** TCP ▪ **[2]** TLS **Note:** When set to 'Not Configured', the value of the parameter SIPTransportType is used.
Web: MWI Subscribe Expiration Time EMS: MWI Expiration Time **[MWIExpirationTime]**	The MWI subscription expiration time in seconds. The default is 7200 seconds. The range is 10 to 2,000,000.
Web: MWI Subscribe Retry Time EMS: Subscribe Retry Time **[SubscribeRetryTime]**	Subscription retry time (in seconds) after last subscription failure. The default is 120 seconds. The range is 10 to 2,000,000.
Web: Subscription Mode **[SubscriptionMode]**	Determines the method the device uses to subscribe to an MWI server. ▪ **[0]** Per Endpoint = Each endpoint subscribes separately - typically used for FXS interfaces (default). ▪ **[1]** Per Gateway = Single subscription for the entire device - typically used for FXO interfaces.
EMS: ETSI VMWI Type One Standard **[ETSIVMWITypeOneStandard]**	Selects the ETSI Visual Message Waiting Indication (VMWI) Type 1 sub-standard. ▪ **[0]** = ETSI VMWI between rings (default) ▪ **[1]** = ETSI VMWI before ring DT_AS ▪ **[2]** = ETSI VMWI before ring RP_AS ▪ **[3]** = ETSI VMWI before ring LR_DT_AS ▪ **[4]** = ETSI VMWI not ring related DT_AS ▪ **[5]** = ETSI VMWI not ring related RP_AS ▪ **[6]** = ETSI VMWI not ring related LR_DT_AS **Note:** For this parameter to take effect, a device reset is required.
EMS: Bellcore VMWI Type One Standard **[BellcoreVMWITypeOneStandard]**	Selects the Bellcore VMWI sub-standard. ▪ **[0]** = Between rings (default). ▪ **[1]** = Not ring related. **Note:** For this parameter to take effect, a device reset is required.

6.8.5 Call Hold Parameters

The call hold parameters are described in the table below.

Table 6-39: Call Hold Parameters

Parameter	Description
Web/EMS: Enable Hold **[EnableHold]**	For digital interfaces: Enables interworking of the Hold/Retrieve supplementary service from PRI to SIP. ▪ **[0]** Disable ▪ **[1]** Enable (default) For analog interfaces: If the Hold service is enabled, a user can place the call on hold (or remove from hold) using the Hook Flash button. On receiving a Hold request, the remote party is placed on hold and hears the hold tone. **Note:** ▪ For digital interfaces: To support interworking of the Hold/Retrieve supplementary service from SIP to ISDN (for QSIG and Euro ISDN), set the parameter EnableHold2ISDN to 1. ▪ For analog interfaces: To use this service, the devices at both ends must support this option.
Web/EMS: Hold Format **[HoldFormat]**	Determines the format of the SDP in the Re-INVITE hold request. ▪ **[0]** 0.0.0.0 = The SDP "c=" field contains the IP address "0.0.0.0" and the "a=inactive" attribute (default). ▪ **[1]** Send Only = The SDP "c=" field contains the device's IP address and the "a=sendonly" attribute. **Notes:** ▪ The <device> does not send any RTP packets when it is in hold state (for both hold formats). ▪ For digital interfaces: This parameter is applicable only to QSIG and Euro ISDN protocols.
Web/EMS:Held Timeout **[HeldTimeout]**	Determines the time interval that the device can allow a call to remain on hold. If a Resume (un-hold Re-INVITE) message is received before the timer expires, the call is renewed. If this timer expires, the call is released. ▪ **[-1]** = The call is placed on hold indefinitely until the initiator of on hold retrieves the call again(default). ▪ **[0 - 2400]** =Time to wait in seconds after which the call is released.
Web: Call Hold Reminder Ring Timeout EMS: CHRR Timeout **[CHRRTimeout]**	Defines the timeout (in seconds) for applying the Call Hold Reminder Ring. If a user hangs up while a call is still on hold, then the FXS interface immediately rings the extension for the duration specified by this parameter. If the user off-hooks the phone, the call becomes active. The valid range is 0 to 600. The default value is 30. **Note:** This parameter is applicable only to FXS interfaces.

6.8.6 Call Transfer Parameters

The call transfer parameters are described in the table below.

Table 6-40: Call Transfer Parameters

Parameter	Description
Web/EMS: Enable Transfer **[EnableTransfer]**	Determines whether call transfer is enabled. ▪ **[0]** Disable = Disable the call transfer service. ▪ **[1]** Enable = The device responds to a REFER message with the Referred-To header to initiate a call transfer(default). For analog interfaces: If the transfer service is enabled, the user can activate Transfer using hook-flash signaling. If this service is enabled, the remote party performs the call transfer. **Notes:** ▪ To use call transfer, the devices at both ends must support this option. ▪ To use call transfer, set the parameter EnableHold to 1.
Web: Transfer Prefix EMS: Logical Prefix For Transferred Call **[xferPrefix]**	Defines the string that is added as a prefix to the transferred/forwarded called number when the REFER/3xx message is received. **Notes:** ▪ The number manipulation rules apply to the user part of the Refer-To and/or Contact URI before it is sent in the INVITE message. ▪ This parameter can be used to apply different manipulation rules to differentiate transferred/forwarded (only for analog interfaces) number from the originally dialed number.
Web: Transfer Prefix IP 2 Tel **[XferPrefixIP2Tel]**	Defines the prefix that is added to the destination number received in the SIP Refer-To header (for IP-to-Tel calls). This parameter is applicable to FXO/CAS Blind Transfer modes (LineTransferMode = 1, 2 or 3 for FXO; TrunkTransferMode = 3 for CAS). The valid range is a string of up to 9 characters. The default is an empty string. **Note:** This parameter is also applicable to ISDN Blind Transfer, according to AT&T Toll Free Transfer Connect Service (TR 50075) “Courtesy Transfer-Human-No Data”. To support this transfer mode, you need to configure the parameter XferPrefixIP2Tel to "*8" and the parameter TrunkTransferMode to 5.
Web/EMS: Enable Semi-Attended Transfer **[EnableSemiAttendedTransfer]**	Determines the device behavior when Transfer is initiated while in Alerting state. ▪ **[0]** Disable = Send REFER with the Replaces header (default). ▪ **[1]** Enable = Send CANCEL, and after a 487 response is received, send REFER without the Replaces header.

Parameter	Description
[KeyBlindTransfer]	Keypad sequence that activates blind transfer for Tel-to-IP calls. There are two possible scenarios: ▪ Option 1: After this sequence is dialed, the current call is put on hold (using Re-INVITE), a dial tone is played to the B-channel, and then phone number collection starts. ▪ Option 2: A Hook-Flash is pressed, the current call is put on hold, a dial tone is played to the B-channel, and then digit collection starts. After this sequence is identified, the device continues the collection of the destination phone number. For both options, after the phone number is collected, it's sent to the transferee in a SIP REFER request (without a Replaces header). The call is then terminated and a confirmation tone is played to the B-channel. If the phone number collection fails due to a mismatch, a reorder tone is played to the B-channel. **Note:** It is possible to configure whether the KeyBlindTransfer code is added as a prefix to the dialed destination number, by using the parameter KeyBlindTransferAddPrefix.
EMS: Blind Transfer Add Prefix **[KeyBlindTransferAddPrefix]**	Determines whether the device adds the Blind Transfer code (KeyBlindTransfer) to the dialed destination number. ▪ **[0]** Disable (default). ▪ **[1]** Enable. **Note:** This parameter is applicable only to FXO and FXS interfaces.
EMS: Blind Transfer Disconnect Timeout **[BlindTransferDisconnectTimeout]**	Defines the duration (in milliseconds) for which the device waits for a disconnection from the Tel side after the Blind Transfer Code (KeyBlindTransfer) has been identified. When this timer expires, a SIP REFER message is sent toward the IP side. If this parameter is set to 0, the REFER message is immediately sent. The valid range is 0 to 1,000,000. The default is 0.

6.8.7 Three-Way Conferencing Parameters

The three-way conferencing parameters are described in the table below.

Table 6-41: Three-Way Conferencing Parameters

Parameter	Description
Web: Enable 3-Way Conference EMS: Enable 3 Way **[Enable3WayConference]**	Enables or disables the 3-Way Conference feature. ▪ **[0]** Disable = Disable (default) ▪ **[1]** Enable = Enables 3-way conferencing **Note:** For this parameter to take effect, a device reset is required.
Web: 3-Way Conference Mode EMS: 3 Way Mode	Defines the mode of operation when the 3-Way Conference feature is used. ▪ **[0]** AudioCodes Media Server = The Conference-initiating INVITE

Parameter	Description
[3WayConferenceMode]	(sent by the device) uses the ConferenceID concatenated with a unique identifier as the Request-URI. This same Request-URI is set as the Refer-To header value in the REFER messages that are sent to the two remote parties. This conference mode is used when operating with AudioCodes IPMedia conferencing server. (Default) ▪ **[1]** Non-AudioCodes Media Server = The Conference-initiating INVITE (sent by the device) uses only the ConferenceID as the Request-URI. The conference server sets the Contact header of the 200 OK response to the actual unique identifier (Conference URI) to be used by the participants. This Conference URI is then included (by the device) in the Refer-To header value in the REFER messages sent by the device to the remote parties. The remote parties join the conference by sending INVITE messages to the conference using this conference URI. **Notes:** ▪ This parameter is applicable only to FXS interfaces. ▪ When using an external conference server (i.e., options **[0]** or **[1]),** more than one three-way conference may be supported (up to six).
Web: Establish Conference Code EMS: Establish Code **[ConferenceCode]**	Defines the digit pattern, which upon detection, generates the Conference-initiating INVITE when 3-way conferencing is enabled (Enable3WayConference is set to 1). The valid range is a 25-character string. The default is "!" (Hook-Flash).
Web/EMS: Conference ID **[ConferenceID]**	Defines the Conference Identification string (up to 16 characters). The default value is 'conf'. For 3-way conferencing using an external media server: The device uses this identifier in the Conference-initiating INVITE that is sent to the media server when Enable3WayConference is set to 1. When using the Mediant 1000 Media Processing Module (MPM): To join a conference, the INVITE URI must include the Conference ID string, preceded by the number of the participants in the conference, and terminated by a unique number. For example: INVITE sip:4MyConference1234@10.1.10.10. INVITE messages with the same URI join the same conference. For example: ConferenceID = MyConference.

6.8.8 Emergency Call Parameters

The emergency call parameters are described in the table below.

Table 6-42: Emergency Call Parameters

Parameter	Description
EMS: Enable 911 PSAP **[Enable911PSAP]**	If enabled, the device supports E911 DID protocol according to Bellcore GR-350-CORE standard. This protocol defines signaling between E911 Tandem Switches and the PSAP, using analog loop-start lines. The FXO device can be installed instead of an E911 switch, connected directly to PSAP DID loop-start lines. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable **Note:** This parameter is applicable only to FXO interfaces.

Parameter	Description
Web/EMS: Emergency Numbers **[EmergencyNumbers]**	Defines a list of numbers, which are defined as 'emergency numbers'. When one of these numbers is dialed, the outgoing INVITE message includes the Priority and Resource-Priority headers. If the user sets the phone on-hook, the call is not disconnected, but instead a Hold Re-INVITE request is sent to the remote party. Only if the remote party disconnects the call (i.e., a BYE is received) or a timer expires (set by the parameter EmergencyRegretTimeout) is the call terminated. The list can include up to four different numbers, where each number can be up to four digits long. Example: EmergencyNumbers = '100','911','112' **Note:** This parameter is applicable only to FXS interfaces.
Web: Emergency Calls Regret Timeout EMS: Emergency Regret Timeout **[EmergencyRegretTimeout]**	Determines the time (in minutes) that the device waits before tearing-down an emergency call (defined by the parameter EmergencyNumbers). Until this time expires, an emergency call can only be disconnected by the remote party, typically, by a Public Safety Answering Point (PSAP). The valid range is 1 to 30. The default value is 10. **Note:** This parameter is applicable only to FXS interfaces.

6.8.9 FXS Call Cut-Through Parameter

The FXS off-hook, call cut-through parameter is described in the table below.

Table 6-43: Call Cut-Through Parameter

Parameter	Description
Web: Enable Calls Cut Through EMS: Cut Through **[CutThrough]**	Enables FXS endpoints to receive incoming IP calls while the port is in an off-hook state. ▪ **[0]** Disable (default). ▪ **[1]** Enable. If enabled, the FXS interface answers the call and 'cuts through' the voice channel if there is no other active call on the port, even if the port is in off-hook state. When the call is terminated (by the remote party), the device plays a reorder tone for a user-defined time (configured by the parameter TimeForReorderTone) and is then ready to answer the next incoming call without on-hooking the phone. The waiting call is automatically answered by the device when the current call is terminated (configured by setting the parameter EnableCallWaiting to 1). **Note:** This feature is applicable only to FXS interfaces.

6.8.10 Automatic Dialing Parameters

The automatic dialing upon off-hook parameters are described in the table below.

Table 6-44: Automatic Dialing Parameters

Parameter	Description
Web: Automatic Dialing Table **EMS: SIP Endpoints > Auto Dial**	
[TargetOfChannel]	This *ini* file table parameter defines telephone numbers that are automatically dialed when a specific FXS or FXO port is used (i.e., telephone is off-hooked). The format of this parameter is as follows: [TargetOfChannel] FORMAT TargetOfChannel_Index = TargetOfChannel_Destination, TargetOfChannel_Type, TargetOfChannel_Module, TargetOfChannel_Port; [\TargetOfChannel] Where, ▪ Destination = Destination phone number that you want dialed. ▪ Type: ✓ **[0]** Disable = automatic dialing is disabled. ✓ **[1]** Enable = Destination phone number is automatically dialed if phone is off-hooked (for FXS interface) or ring signal is applied to port (FXO interface). ✓ **[2]** Hotline = enables the Hotline feature where if the phone is off-hooked and no digit is pressed for a user-defined duration (configured by the parameter HotLineToneDuration), the destination phone number is automatically dialed. ▪ Module = Module number (where 1 depicts the module in Slot 1). ▪ Port = Port number (where 1 depicts the Port 1 of the module). For example, the below configuration defines automatic dialing of phone number 911 when the phone that is connected to Port 1 of Module 1 is off-hooked: TargetOfChannel 0 = 911,1,1,1; (phone number "1002" is automatically dialed for Port 1 of Module 1) **Notes:** ▪ This is parameter is applicable to FXS and FXO interfaces. ▪ The indexing of this *ini* file table parameter starts at 0. ▪ Define this parameter for each device port that implements Automatic Dialing. ▪ After a ring signal is detected on an 'Enabled' FXO port, the device initiates a call to the destination number without seizing the line. The line is seized only after the call is answered. After a ring signal is detected on a 'Disabled' or 'Hotline' FXO port, the device seizes the line. ▪ For configuring this table using the Web interface, refer to "Configuring Automatic Dialing" on page 172. ▪ For an explanation on using *ini* file table parameters, refer to "Configuring *ini* File Table Parameters" on page 222.

6.8.11 Direct Inward Dialing Parameters

The Direct Inward Dialing (DID) parameters are described in the table below.

Table 6-45: DID Parameters

Parameter	Description
Web/EMS: Enable DID Wink **[EnableDIDWink]**	Enables Direct Inward Dialing (DID) using Wink-Start signaling. ▪ **[0]** Disable = Disables DID Wink(default). ▪ **[1]** Enable = Enables DID Wink. If enabled, the device can be used for connection to EIA/TIA-464B DID Loop Start lines. Both FXO (detection) and FXS (generation) are supported. An FXO interface dials DTMF digits after a Wink signal is detected (instead of a Dial tone). An FXS interface generates the Wink signal after the detection of off-hook (instead of playing a Dial tone).
Web/EMS: Delay Before DID Wink **[DelayBeforeDIDWink]**	Defines the time interval (in msec) between detection of off-hook and generation of a DID Wink. The valid range is 0 to 1,000. The default value is 0. **Note:** This parameters is applicable only to FXS interfaces.
EMS: NTT DID Signalling Form **[NTTDIDSignallingForm]**	Determines the type of DID signaling support for NTT (Japan) modem: DTMF- or Frequency Shift Keying (FSK)-based signaling. The devices can be connected to Japan's NTT PBX using 'Modem' DID lines. These DID lines are used to deliver a called number to the PBX. ▪ **[0]** = FSK-based signaling (default) ▪ **[1]** = DTMF-based signaling **Note:** This parameter is applicable only to FXS interfaces.
EMS: Enable DID **[EnableDID]**	This *ini* file table parameter enables support for Japan NTT 'Modem' DID. FXS interfaces can be connected to Japan's NTT PBX using 'Modem' DID lines. These DID lines are used to deliver a called number to the PBX. The DID signal can be sent alone or combined with an NTT Caller ID signal. The format of this parameter is as follows: [EnableDID] FORMAT EnableDID_Index = EnableDID_IsEnable, EnableDID_Port, EnableDID_Module; [\EnableDID] Where, ▪ IsEnable = Enables **[1]** or disables **[0]** (default) Japan NTT Modem DID support. ▪ Port = Port number. ▪ Module = Module number. For example: [EnableDID] EnableDID 0 = 1,1,2; (DID is enabled on Port 1 of Module 2) [\EnableDID] **Notes:** ▪ This parameter is applicable only to FXS interfaces. ▪ For an explanation on using *ini* file table parameters, refer to

Parameter	Description
	"Configuring *ini* File Table Parameters" on page 222.
[WinkTime]	Defines the time (in msec) elapsed between two consecutive polarity reversals. This parameter can be used for DID signaling, for example, E911 lines to the Public Safety Answering Point (PSAP), according to the Bellcore GR-350-CORE standard (refer to the *ini* file parameter Enable911PSAP). The valid range is 0 to 4,294,967,295. The default is 200. **Notes:** ▪ This parameter is applicable to FXS and FXO interfaces. ▪ For this parameter to take effect, a device reset is required.

6.8.12 MLPP Parameters

The Multilevel Precedence and Preemption (MLPP) parameters are described in the table below.

Table 6-46: MLPP Parameters

Parameter	Description
Web/EMS: Call Priority Mode **[CallPriorityMode]**	Enables MLPP Priority Call handling. ▪ **[0]** Disable = Disable (default). ▪ **[1]** MLPP = Priority Calls handling is enabled.
Web: MLPP Default Namespace EMS: Default Name Space **[MLPPDefaultNamespace]**	Determines the Namespace used for MLPP calls received from the ISDN side and destined for the Application server. The Namespace value is not present in the Precedence IE of the PRI Setup message. Therefore, the value is used in the Resource-Priority header of the outgoing SIP INVITE request. ▪ **[1]** DSN = DSN (default) ▪ **[2]** DOD = DOD ▪ **[3]** DRSN = DRSN
Web/EMS: Default Call Priority **[SIPDefaultCallPriority]**	Defines the default call priority for MLPP calls. ▪ **[0]** 0 = ROUTINE (default) ▪ **[2]** 2 = PRIORITY ▪ **[4]** 4 = IMMEDIATE ▪ **[6]** 6 = FLASH ▪ **[8]** 8 = FLASH-OVERRIDE ▪ **[9]** 9 = FLASH-OVERRIDE-OVERRIDE If the incoming SIP INVITE request doesn't contain a valid priority value in the SIP Resource-Priority header, the default value is used in the Precedence IE (after translation to the relevant ISDN Precedence value) of the outgoing PRI Setup message. If the incoming PRI Setup message doesn't contain a valid Precedence Level value, the default value is used in the Resource-Priority header of the outgoing SIP INVITE request. In this scenario, the character string is sent without translation to a numerical value.

Parameter	Description
Web: MLPP DiffServ EMS: Diff Serv **[MLPPDiffserv]**	Defines the DiffServ value (differentiated services code point/DSCP) used in IP packets containing SIP messages that are related to MLPP calls. This parameter defines DiffServ for incoming and outgoing MLPP calls with the Resource-Priority header. The valid range is 0 to 63. The default value is 50.
Web/EMS: Preemption Tone Duration **[PreemptionToneDuration]**	Defines the duration (in seconds) in which the device plays a preemption tone to the Tel and IP sides if a call is preempted. The valid range is 0 to 60. The default is 3. **Note:** If set to 0, no preemption tone is played.
Web: MLPP Normalized Service Domain EMS: Normalized Service Domain **[MLPPNormalizedServiceDomain]**	MLPP normalized service domain string. If the device receives an MLPP ISDN incoming call, it uses the parameter (if different from 'FFFFFF') as a Service domain in the SIP Resource-Priority header in outgoing INVITE messages. If the parameter is configured to 'FFFFFF', the Resource-Priority header is set to the MLPP Service Domain obtained from the Precedence IE. The valid value is a 6 hexadecimal digits. The default is '000000'. **Note:** This parameter is applicable only to the MLPP NI-2 ISDN variant with CallPriorityMode set to 1.
[MLPPNetworkIdentifier]	Defines the MLPP network identifier (i.e., International prefix or Telephone Country Code/TCC) for IP-to-ISDN calls, according to the UCR 2008 and ITU Q.955 specifications. The valid range is 1 to 999. The default is 1 (i.e., USA). The MLPP network identifier is sent in the Facility IE of the ISDN Setup message. For example: ▪ MLPPNetworkIdentifier set to default (i.e., USA, 1): PlaceCall- MLPPNetworkID:0100 MlppServiceDomain:123abc, MlppPrecLevel:5 Fac(1c): 91 a1 15 02 01 05 02 01 19 30 0d 0a 01 05 0a 01 01 04 05 01 00 12 3a bc ▪ MLPPNetworkIdentifier set to 490: PlaceCall- MLPPNetworkID:9004 MlppServiceDomain:123abc, MlppPrecLevel:5 Fac(1c): 91 a1 15 02 01 0a 02 01 19 30 0d 0a 01 05 0a 01 01 04 05 90 04 12 3a bc
Web: MLPP Default Service Domain EMS: Default Service Domain **[MLPPDefaultServiceDomain]**	MLPP default service domain string. If the device receives a non-MLPP ISDN incoming call (without a Precedence IE), it uses the parameter (if different than "FFFFFF") as a Service domain in the SIP Resource-Priority header in outgoing (Tel-to-IP calls) INVITE messages. This parameter is used in conjunction with the parameter SIPDefaultCallPriority. If MLPPDefaultServiceDomain is set to 'FFFFFF', the device interworks the non-MLPP ISDN call to non-MLPP SIP call, and the outgoing INVITE does not contain the Resource-Priority header. The valid value is a 6 hexadecimal digits. The default is

Parameter	Description
	"000000". **Note:** This parameter is applicable only to the MLPP NI-2 ISDN variant with CallPriorityMode set to 1.
EMS: E911 MLPP Behavior **[E911MLPPBehavior]**	Defines the E911(or Emergency Telecommunication Services - ETS) MLPP Preemption mode: ▪ **[0]** Standard Mode - ETS calls have the highest priority and preempt any MLPP call (default). ▪ **[1]** Treat as routine mode - ETS calls are handled as routine calls. **Note:** This parameter is applicable only to analog interfaces.
Web/EMS: Precedence Ringing Type **[PrecedenceRingingType]**	Defines the index of the Precedence Ringing tone in the Call Progress Tones (CPT) file. This tone is used when the parameter CallPriorityMode is set to 1 and a Precedence call is received from the IP side. The valid range is -1 to 16. The default value is -1 (i.e., plays standard Ringing tone). **Note:** This parameter is applicable only to analog interfaces.
Multiple Differentiated Services Code Points (DSCP) per MLPP Call Priority Level (Precedence) Parameters The MLPP service allows placement of priority calls, where properly validated users can preempt (terminate) lower-priority phone calls with higher-priority calls. For each MLPP call priority level, the DSCP can be set to a value from 0 to 63. The Resource Priority value in the Resource-Priority SIP header can be one of the following: **MLPP Precedence Level** — **Precedence Level in Resource-Priority SIP Header** 0 (lowest) — routine 2 — priority 4 — immediate 6 — flash 8 — flash-override 9 (highest) — flash-override-override	
Web/EMS: RTP DSCP for MLPP Routine **[MLPPRoutineRTPDSCP]**	Defines the RTP DSCP for MLPP Routine precedence call level. The valid range is -1 to 63. The default is -1. **Note:** If set to -1, the DiffServ value is taken from the global parameter PremiumServiceClassMediaDiffServ or as defined for IP Profiles per call (using the parameter IPProfile).
Web/EMS: RTP DSCP for MLPP Priority **[MLPPPriorityRTPDSCP]**	Defines the RTP DSCP for MLPP Priority precedence call level. The valid range is -1 to 63. The default is -1. **Note:** If set to -1, the DiffServ value is taken from the global parameter PremiumServiceClassMediaDiffServ or as defined for IP Profiles per call (using the parameter IPProfile).

Parameter	Description
Web/EMS: RTP DSCP for MLPP Immediate **[MLPPImmediateRTPDSCP]**	Defines the RTP DSCP for MLPP Immediate precedence call level. The valid range is -1 to 63. The default is -1. **Note:** If set to -1, the DiffServ value is taken from the global parameter PremiumServiceClassMediaDiffServ or as defined for IP Profiles per call (using the parameter IPProfile).
Web/EMS: RTP DSCP for MLPP Flash **[MLPPFlashRTPDSCP]**	Defines the RTP DSCP for MLPP Flash precedence call level. The valid range is -1 to 63. The default is -1. **Note:** If set to -1, the DiffServ value is taken from the global parameter PremiumServiceClassMediaDiffServ or as defined for IP Profiles per call (using the parameter IPProfile).
Web/EMS: RTP DSCP for MLPP Flash Override **[MLPPFlashOverRTPDSCP]**	Defines the RTP DSCP for MLPP Flash-Override precedence call level. The valid range is -1 to 63. The default is -1. **Note:** If set to -1, the DiffServ value is taken from the global parameter PremiumServiceClassMediaDiffServ or as defined for IP Profiles per call (using the parameter IPProfile).
Web/EMS: RTP DSCP for MLPP Flash-Override-Override **[MLPPFlashOverOverRTPDSCP]**	Defines the RTP DSCP for MLPP Flash-Override-Override precedence call level. The valid range is -1 to 63. The default is -1. **Note:** If set to -1, the DiffServ value is taken from the global parameter PremiumServiceClassMediaDiffServ or as defined for IP Profiles per call (using the parameter IPProfile).

6.9 Standalone Survivability Parameters

The Stand-alone Survivability (SAS) parameters are described in the table below.

Table 6-47: SAS Parameters

Parameter	Description
Web: Enable SAS EMS: Enable **[EnableSAS]**	Enables the Stand-Alone Survivability (SAS) feature. ▪ **[0]** Disable Disabled (default) ▪ **[1]** Enable = SAS is enabled When enabled, the device receives the registration requests from different SIP entities in the local network and then forwards them to the defined proxy. If the connection to the proxy fails ('Emergency Mode'), the device serves as a proxy by allowing calls internal to the local network or outgoing to PSTN. **Note:** For this parameter to take effect, a device reset is required.
Web: SAS Local SIP UDP Port EMS: Local SIP UDP **[SASLocalSIPUDPPort]**	Local UDP port for sending and receiving SIP messages for SAS. The SIP entities in the local network need to send the registration requests to this port. When forwarding the requests to the proxy ('Normal Mode'), this port serves as the source port. The valid range is 1 to 65,534. The default value is 5080.

Parameter	Description
Web: SAS Default Gateway IP EMS: Default Gateway IP **[SASDefaultGatewayIP]**	The default gateway used in SAS 'Emergency Mode'. When an incoming SIP INVITE is received and the destination Address-Of-Record is not included in the SAS database, the request is immediately sent to this default gateway. The address can be configured as an IP address (dotted-decimal notation) or as a domain name (up to 49 characters). The default is a null string, which is interpreted as the local IP address of the gateway.
Web: SAS Registration Time EMS: Registration Time **[SASRegistrationTime]**	Determines the value of the SIP Expires header that is sent in a 200 OK response to an incoming REGISTER message when in SAS 'Emergency Mode'. The valid range is 0 (Analog) or 10 (Digital) to 2,000,000. The default value is 20.
Web: SAS Local SIP TCP Port EMS: Local SIP TCP Port **[SASLocalSIPTCPPort]**	Local TCP port used to send/receive SIP messages for the SAS application. The SIP entities in the local network need to send the registration requests to this port. When forwarding the requests to the proxy ('Normal Mode'), this port serves as the source port. The valid range is 1 to 65,534. The default value is 5080.
Web: SAS Local SIP TLS Port EMS: Local SIP TLS Port **[SASLocalSIPTLSPort]**	Local TLS port used to send/receive SIP messages for the SAS application. The SIP entities in the local network need to send the registration requests to this port. When forwarding the requests to the proxy ('Normal Mode'), this port serves as the source port. The valid range is 1 to 65,534. The default value is 5081.
Web/EMS: Enable Record-Route **[SASEnableRecordRoute]**	Determines whether the device's SAS application adds the SIP Record-Route header to SIP requests. This ensures that SIP messages traverse the device's SAS agent by including the SAS IP address in the Record-Route header. ▪ **[0]** Disable (default) ▪ **[1]** Enable The Record-Route header is inserted in a request by a SAS proxy to force future requests in the dialog session to be routed through the SAS agent. Each traversed proxy in the path can insert this header, causing all future dialogs in the session to pass through it as well. When this feature is enabled, the SIP Record-Route header includes the URI "lr" parameter. The presence of this parameter indicates loose routing; the lack of 'lt' indicates strict routing. For example: ▪ Loose routing: Record-Route: <sip:server10.biloxi.com;lr> ▪ Strict routing: Record-Route: <sip:bigbox3.site3.atlanta.com>
Web: SAS Proxy Set EMS: Proxy Set **[SASProxySet]**	Determines the Proxy Set (index number) used in SAS Normal mode to forward REGISTER and INVITE requests from the users that are served by the SAS application. The valid range is 0 to 5. The default value is 0 (i.e., default Proxy Set).

Parameter	Description
Web: Redundant SAS Proxy Set EMS: Redundant Proxy Set **[RedundantSASProxySet]**	Determines the Proxy Set (index number) used in SAS Emergency mode for fallback when the user is not found in the Registered Users database. Each time a new SIP request arrives, the SAS application checks whether the user is listed in the registration database. If the user is located in the database, the request is sent to the user. If the user is not found, the request is forwarded to the next redundant SAS defined in the Redundant SAS Proxy Set. If that SAS Proxy IP appears in the Via header of the request, it is not forwarded (thereby, preventing loops in the request's course). If no such redundant SAS exists, the SAS sends the request to its default gateway (configured by the parameter SASDefaultGatewayIP). The valid range is -1 to 5. The default value is -1 (i.e., no redundant Proxy Set).
[SASEnableContactReplace]	Enables the device to change the SIP Contact header so that it points to the SAS host and therefore, the top-most SIP Via header and the Contact header point to the same host. ▪ **[0]** (default) = Disable - when relaying requests, the SAS agent adds a new Via header (with the SAS IP address) as the top-most Via header and retains the original Contact header. Thus, the top-most Via header and the Contact header point to different hosts. ▪ **[1]** = Enable - the device changes the Contact header so that it points to the SAS host and therefore, the top-most Via header and the Contact header point to the same host. **Note:** Operating in this mode causes all incoming dialog requests to traverse the SAS, which may cause load problems.
Web: SAS Survivability Mode EMS: Survivability Mode **[SASSurvivabilityMode]**	Determines the Survivability mode used by the SAS application. ▪ **[0]** Standard = All incoming INVITE and REGISTER requests are forwarded to the defined Proxy list of SASProxySet in Normal mode and handled by the SAS application in Emergency mode (default). ▪ **[1]** Always Emergency = The SAS application does not use Keep-Alive messages towards the SASProxySet, instead it always operates in Emergency mode (as if no Proxy in the SASProxySet is available). ▪ **[2]** Ignore Register = Use regular SAS Normal/Emergency logic (same as option **[0]),** but when in Normal mode incoming REGISTER requests are ignored. ▪ **[3]** Auto-answer REGISTER = When in Normal mode, the device responds to received REGISTER requests by sending a SIP 200 OK (instead of relaying the registration requests to a Proxy), and enters the registrations in its SAS database.
Web: SAS Binding Mode EMS: Binding Mode **[SASBindingMode]**	Determines the SAS application database binding mode. ▪ **[0]** URI = If the incoming AoR in the INVITE requests is using a 'tel:' URI or 'user=phone' is defined, the binding is performed according to the user part of the URI only. Otherwise, the binding is according to the entire URI, i.e., User@Host (default). ▪ **[1]** User Part only = The binding is always performed according to the User Part only.

Parameter	Description
Web: SAS Emergency Numbers **[SASEmergencyNumbers]**	Defines emergency numbers for the device's SAS application. When the device's SAS agent receives a SIP INVITE (from an IP phone) that includes one of the emergency numbers (in the SIP user part), it forwards the INVITE to the default gateway (configured by the parameter SASDefaultGatewayIP), i.e., the device itself, which sends the call directly to the PSTN. This is important for routing emergency numbers such as 911 (in North America) directly to the PSTN. This is applicable to SAS operating in Normal and Emergency modes. Up to four emergency numbers can be defined, where each number can be up to four digits.
[SASEmergencyPrefix]	Defines a prefix that is added to the Request-URI user part of the INVITE message that is sent by the device's SAS agent when in Emergency mode to the default gateway or to any other destination (using the 'IP2IP Routing' table). This parameter is required to differentiate between normal SAS calls routed to the default gateway and emergency SAS calls. Therefore, this allows you to define different manipulation rules for normal and emergency calls. This valid value is a character string. The default is an empty string "".
Web: SAS Registration Manipulation Table **EMS: Stand-Alone Survivability**	
[SASRegistrationManipulation]	This *ini* file table parameter configures the SAS Registration Manipulation table. This table is used by the SAS application to manipulate the user part of an incoming REGISTER request AoR (the To header), before saving it to the registered users database. The format of this table parameter is as follows: [SASRegistrationManipulation] FORMAT SASRegistrationManipulation_Index = SASRegistrationManipulation_RemoveFromRight, SASRegistrationManipulation_LeaveFromRight; [\SASRegistrationManipulation] ▪ RemoveFromRight = number of digits removed from the right side of the user part before saving to the registered user database. ▪ LeaveFromRight = number of digits to keep from the right side. If both RemoveFromRight and LeaveFromRight are defined, the RemoveFromRight is applied first. The registered database contains the AoR before and after the manipulation. The range of both RemoveFromRight and LeaveFromRight is 0 to 30. **Note:** This table can include only one index entry.

Parameter	Description
Web: SAS IP-to-IP Routing Table	
[IP2IPRouting]	This *ini* file table parameter configures the IP-to-IP Routing table for SAS routing rules. The format of this parameter is as follows: [IP2IPRouting] FORMAT IP2IPRouting_Index = IP2IPRouting_SrcIPGroupID, IP2IPRouting_SrcUsernamePrefix, IP2IPRouting_SrcHost, IP2IPRouting_DestUsernamePrefix, IP2IPRouting_DestHost, IP2IPRouting_DestType, IP2IPRouting_DestIPGroupID, IP2IPRouting_DestSRDID, IP2IPRouting_DestAddress, IP2IPRouting_DestPort, IP2IPRouting_DestTransportType, IP2IPRouting_AltRouteOptions; [\IP2IPRouting] For example: IP2IPRouting 1 = -1, *, *, *, *, 0, -1, -1, , 0, -1, 0; **Notes:** ▪ This table can include up to 120 indices (where 0 is the first index). ▪ The parameters SrcIPGroupID, DestSRDID, and AltRouteOptions are not applicable. ▪ For a detailed description of the individual parameters in this table and for configuring this table using the Web interface, refer to "Configuring the IP2IP Routing Table (SAS)" on page 181. ▪ For a description on configuring *ini* file table parameters, refer to "Configuring *ini* File Table Parameters" on page 222.

6.10 IP Media Parameters

The IP media parameters are described in the table below.

Table 6-48: IP Media Parameters

Parameter	Description
Web: Number of Media Channels EMS: Media Channels **[MediaChannels]**	Determines the number of DSP channels that are allocated for IP conferencing, IP streaming, and IP transcoding (other DSP channels can be used for PSTN device). This parameter also determines the number of DSP channels allocated for IP-to-IP sessions (other DSP channels can be used for PSTN interface). Currently, the RTP streams for IP-to-IP calls always transverse through the device, and two DSP channels are allocated per IP-to-IP session. Therefore, the maximum number of supported Media channels for IP-to-IP calls is 120, corresponding to 60 IP-to-IP calls. For a description on DSP utilization, refer to DSP Channel DSP Channel Resources for IP-to-IP Routing. The maximum value of Media Channels depends on the number of installed Media Processing modules (MPM): 1 module = 20 channels; 2 modules = 60; 3 modules = 100. The default value is 0.

Parameter	Description
	Note: For this parameter to take effect, a device reset is required.
[EnableIPMediaChannels]	Determines whether to enable IP media channel support. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable **Notes:** ▪ This parameter is applicable only to Mediant 1000. ▪ For this parameter to take effect, a device reset is required.
[IPmediaChannels]	This *ini* file parameter table defines the number of DSP channels that are "borrowed" from each of the device's digital modules for IP media functionality. The format of this parameter is as follows: [IPMediaChannels] FORMAT IPMediaChannels_Index = IPMediaChannels_ModuleID, IPMediaChannels_DSPChannelsReserved; [\IPMediaChannels] For example, the below settings use 15 and 10 DSP channels from modules 1 and 2, respectively: IPMediaChannels 1 = 1, 15; IPMediaChannels 2 = 2, 10; **Notes:** ▪ This parameter is applicable only to Mediant 1000. ▪ The value of DSPChannelsReserved must be in multiples of 5 (since the reservation is done per DSP device and not per DSP channel). ▪ By default, the MPM module is set to the maximum value of IPM channels, therefore, there is no need to define it. ▪ By default, a PRI module is set to 0 IPM channels. ▪ For an explanation on DSP utilization options, refer to DSP Channel DSP Channel Resources for IP-to-IP Routing.
Web: Enable Voice Streaming **[EnableVoiceStreaming]**	Enables/disables the HTTP Voice Streaming application (play/record). ▪ **[0]** Disable (default). ▪ **[1]** Enable. **Note:** For this parameter to take effect, a device reset is required.
[VoiceStreamUploadMethod]	Defines the HTTP request type for loading the voice stream to the file server. ▪ **[0]** = POST (default). ▪ **[1]** = PUT. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ This parameter is applicable only to MSCML recording.

Parameter	Description
[VoiceStreamUploadPostURI]	Defines the URI used on the POST request to upload voice data from the media server to a Web server. **Note:** For this parameter to take effect, a device reset is required.
[APSEnabled]	Indicates whether Voice Prompt index references refer to audio provided by the Audio Provisioning Server (APS) or by the local Voice Prompts file. ▪ **[0]** = APS disabled. Local Voice Prompts file is used. An audio reference in a play request (such as http://localhost/0) indicates that the Voice Prompt at index 0 in the Voice Prompts file is played. ▪ **[1]** = APS enabled (default). An audio reference (such as http://localhost/99) indicates that the audio segment provisioned on the APS with segment ID 99 is played. **Note:** For this parameter to take effect, a device reset is required.
Web: Calling Number Playback ID **[CallingNumberPlayBackID]**	The Calling Number identification string for local, audio playing of the calling number. When the device receives from the Application Server (or SIP user Agent) a regular SIP INVITE message with a SIP URI that includes this user-defined Calling Number identification string, the device plays the calling number to the phone. For example, upon the receipt of the below INVITE message, the device plays the numbers 1, 0, and then 1: INVITE sip:callingnumber@domain.com; From: <sip:101@10.132.11.245>; The valid value can be up to 16 characters. The default is "callingnumber". **Note:** The APS server support must be enabled to support this feature. Below are the relevant *ini* file parameter settings: ▪ CallingNumberPlayBackID = callingnumber ▪ VpFileUrl = 'http://10.132.10.46/vp.dat' ▪ APSSegmentsFileURL = 'http://10.132.10.46/segments.xml' ▪ APSEnabled = 1 ▪ AMSProfile = 1 ▪ AASPackagesProfile = 3 ▪ EnableVoiceStreaming = 1
Web: NetAnn Announcement ID **[NetAnnAnncID]**	NetAnn identification string (up to 16 characters) for playing an announcement using the NetAnn interface. The application server sends a regular SIP INVITE message with a SIP URI that includes this identifier string and a “play=” parameter that identifies the necessary announcement. The default value is ‘annc’. Example 1: INVITE sip: annc@10.2.3.4;play=http://localhost/1. Example 2: INVITE sip: annc@10.2.3.4;play=http://10.2.3.4/Annc/hello.wav.

Parameter	Description
Web: MSCML ID **[MSCMLID]**	Media Server Control Markup Language (MSCML) identification string (up to 16 characters). To start an MSCML session, the application server sends a regular SIP INVITE message with a SIP URI that includes this string. The default value is 'ivr'. For example: INVITE sip:ivr@10.2.3.4 Subsequent INFO messages carry the requests and responses.
Web: Transcoding ID **[TranscodingID]**	Transcoding identification string (up to 16 characters) used for identifying an incoming Transcoding call. The default value is 'trans'. For detailed information on Transcoding, refer to NetAnn Interface on page 656.
AMS Parameters	
[AmsProfile]	Must be set to 1 to use advanced audio. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable **Note:** For this parameter to take effect, a device reset is required.
[AASPackagesProfile]	Must be set to 3 to use advanced audio. **Note:** For this parameter to take effect, a device reset is required.
[AmsPrimaryLanguage]	Determines the primary language used in the advanced audio package. The default value is "eng". The languages are according to ISO standard 639-2 language codes.
[AmsSecondaryLanguage]	Determines the secondary language used in the advanced audio package. The default value is "heb". The languages are according to ISO standard 639-2 language codes.
[AMSAllowUrlAsAlias]	Determines whether or not play requests for remote URLs are first verified with local audio segments to determine if any have an alias matching for the URL. If a match is found, the corresponding local audio segment is played. ▪ **[0]** = Always use remote storage (default).\ ▪ **[1]** = Check local storage first. One application for this capability is that of a 'provisioned' cache within the device. For details on provisioning an alias and other audio provisioning capabilities, refer to the Audio Provisioning Server (APS) User's Manual.

Parameter	Description
Conferencing Parameters	
Web/EMS: Conference ID **[ConferenceID]**	Conference Identification string (up to 16 characters). The default value is 'conf'. For example: ConferenceID = MyConference **Note:** To join a conference, the INVITE URI must include the Conference ID string, preceded by the number of the participants in the conference, and terminated by a unique number. For example: Invite sip:4MyConference1234@10.1.10.10. INVITE messages with the same URI join the same conference.
Web: Beep on Conference **[BipOnConference]**	Determines whether or not a beep is played when a participant joins or leaves a conference (in the latter case, a beep of a different pitch is heard). ▪ **[0]** Disable = Beep is disabled. ▪ **[1]** Enable = Beep is enabled (default).
Web: Enable Conference DTMF Clamping **[EnableConferenceDTMFClamp]**	Determines the device logic once a DTMF is received on any conference participant. If enabled, the DTMF is not regenerated toward the other conference participants. This logic is only relevant for simple conferencing (NetAnn). ▪ **[0]** Disable = Disable ▪ **[1]** Enable = Enable (default)
Web: Enable Conference DTMF Reporting **[EnableConferenceDTMFReporting]**	Determines the device logic once a DTMF is received on any conference participant. If enabled, the device reports this DTMF in an out-of-band SIP message (according to TxDTMFOptions). This logic is only relevant for simple conferencing (NetAnn). ▪ **[0]** Disable = Disable (default) ▪ **[1]** Enable = Enable
Web: Active Speakers Min. Interval **[ActiveSpeakersNotificationMinInterval]**	Minimum interval (in 100 msec units) between each Active Speaker Notification (ASN) events report. These events report on the active speakers in a conference. The event is issued whenever the active speakers change. Minimum configurable interval between events is 500 msec (5 units). The range is 5 to 2147483647 units. The default is 20 (i.e., 100 msec).
Web: Playback Audio Format **[cpPlayCoder]**	Determines the coder when playing a RAW file. ▪ **[1]** G711 Mulaw ▪ **[2]** G711 Alaw (default)
Web: Record Audio Format **[cpRecordCoder]**	Determines the coder for recording all supported file types. ▪ **[1]** G711 Mulaw ▪ **[2]** G711 Alaw (default) **Note:** For this parameter to take effect, a device reset is required.
Web: End of Record Trim **[cpEndOfRecordCutTime]**	The maximum amount (in milliseconds) of audio to remove from the end of a recording. This is used to remove the DTMF signals generated by the end user for terminating the record. The valid range is 0 to 65,535. The default is 0.

Parameter	Description
Automatic Gain Control (AGC) Parameters	
Web: Enable AGC EMS: AGC Enable **[EnableAGC]**	Activates the AGC mechanism. The AGC mechanism adjusts the level of the received signal to maintain a steady (configurable) volume level. ▪ **[0]** Disable (default). ▪ **[1]** Enable. **Note:** For a description of AGC, refer to Automatic Gain Control (AGC) on page 634.
Web: AGC Slope EMS: Gain Slope **[AGCGainSlope]**	Determines the AGC convergence rate: ▪ **[0]** 0 = 0.25 dB/sec ▪ **[1]** 1 = 0.50 dB/sec ▪ **[2]** 2 = 0.75 dB/sec ▪ **[3]** 3 = 1.00 dB/sec (default) ▪ **[4]** 4 = 1.25 dB/sec ▪ **[5]** 5 = 1.50 dB/sec ▪ **[6]** 6 = 1.75 dB/sec ▪ **[7]** 7 = 2.00 dB/sec ▪ **[8]** 8 = 2.50 dB/sec ▪ **[9]** 9 = 3.00 dB/sec ▪ **[10]** 10 = 3.50 dB/sec ▪ **[11]** 11 = 4.00 dB/sec ▪ **[12]** 12 = 4.50 dB/sec ▪ **[13]** 13 = 5.00 dB/sec ▪ **[14]** 14 = 5.50 dB/sec ▪ **[15]** 15 = 6.00 dB/sec ▪ **[16]** 16 = 7.00 dB/sec ▪ **[17]** 17 = 8.00 dB/sec ▪ **[18]** 18 = 9.00 dB/sec ▪ **[19]** 19 = 10.00 dB/sec ▪ **[20]** 20 = 11.00 dB/sec ▪ **[21]** 21 = 12.00 dB/sec ▪ **[22]** 22 = 13.00 dB/sec ▪ **[23]** 23 = 14.00 dB/sec ▪ **[24]** 24 = 15.00 dB/sec ▪ **[25]** 25 = 20.00 dB/sec ▪ **[26]** 26 = 25.00 dB/sec ▪ **[27]** 27 = 30.00 dB/sec ▪ **[28]** 28 = 35.00 dB/sec ▪ **[29]** 29 = 40.00 dB/sec ▪ **[30]** 30 = 50.00 dB/sec ▪ **[31]** 31 = 70.00 dB/sec

Parameter	Description
Web: AGC Redirection EMS: Redirection **[AGCRedirection]**	Determines the AGC direction. ▪ **[0]** 0 = AGC works on signals from the TDM side (default). ▪ **[1]** 1 = AGC works on signals from the IP side.
Web: AGC Target Energy EMS: Target Energy **[AGCTargetEnergy]**	Determines the signal energy value (dBm) that the AGC attempts to attain. The valid range is 0 to -63 dBm. The default value is -19 dBm.
EMS: Minimal Gain **[AGCMinGain]**	Defines the minimum gain (in dB) by the AGC when activated. The range is 0 to -31. The default is -20. **Note:** For this parameter to take effect, a device reset is required.
EMS: Maximal Gain **[AGCMaxGain]**	Defines the maximum gain (in dB) by the AGC when activated. The range is 0 to 18. The default is 15. **Note:** For this parameter to take effect, a device reset is required.
EMS: Disable Fast Adaptation **[AGCDisableFastAdaptation]**	Disables the AGC Fast Adaptation mode. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable **Note:** For this parameter to take effect, a device reset is required.
Answer Machine Detector (AMD) Parameters	
[AMDMinimumVoiceLength]	Determines the AMD minimum voice activity detection (in 5-ms units). Voice activity duration below this threshold is ignored and considered as non-voice. The valid value range is 10 to 100. The default is 42 (i.e., 210 ms).
Web: Answer Machine Detector Sensitivity Resolution EMS: Sensitivity Resolution **[AMDSensitivityResolution]**	Determines the AMD detection sensitivity resolution (normal or high). ▪ **[0]** Normal (default) = Normal detection sensitivity resolution (8 sensitivity levels), configured by the parameter AMDDetectionSensitivity. ▪ **[1]** High = High detection sensitivity resolution (16 sensitivity levels), configured by the parameter AMDDetectionSensitivityHighResolution.
Web: Answer Machine Detector Sensitivity EMS: Sensitivity **[AMDDetectionSensitivity]**	Determines the AMD detection sensitivity. AMD can be useful in automatic dialing applications. In some of these applications, it is important to detect if a human voice or an answering machine is answering the call. AMD can be activated and de-activated only after a channel is already open. The direction of the detection (PSTN or IP) can also be configured (using the parameter AMDDetectionDirection). This parameter is used if the parameter AMDSensitivityResolution is set to 0 ('Normal'). The valid value range is 0 to 7, where 0 is the best detection for answering machines and 7 is the best detection for live calls (i.e., voice detection). The default is 3. For a detailed description of AMD, refer to Answer Machine

Parameter	Description
	Detector (AMD) on page 589. **Notes:** ▪ For configuring higher sensitivity resolutions (i.e., greater than 7), set the parameter AMDSensitivityResolution to 1 (High), and then for *ini* file configuration use the parameter AMDDetectionSensitivityHighResolution to define the sensitivity level. For Web interface configuration, use the AMDDetectionSensitivity parameter. ▪ To enable the AMD feature, set the *ini* file parameter EnableDSPIPMDetectors to 1.
EMS: Detection Sensitivity High Resolution **[AMDDetectionSensitivityHighResolution]**	Determines the AMD high-resolution detection sensitivity. The high resolution has 16 levels of sensitivity (while the normal resolution, configured by the parameter AMDDetectionSensitivity has only 8 levels). The valid value range is 0 (for best detection of an answering machine) to 15 (for best detection of a live call). The default value is 8. **Note:** For configuring high sensitivity resolution in the Web interface, use the Web parameter that corresponds to the *ini* file parameter AMDDetectionSensitivity.
EMS: Time Out **[AMDTimeout]**	Timeout (in msec) between receiving Connect messages from the ISDN and sending AMD results. The valid range is 1 to 30,000. The default is 2,000 (i.e., 2 seconds).
EMS: Detection Direction **[AMDDetectionDirection]**	Determines the AMD detection direction. ▪ **[0]** = Detection from the PSTN side (default) ▪ **[1]** = Detection from the IP side
Web/EMS: AMD Beep Detection Mode **[AMDBeepDetectionMode]**	Determines the AMD beep detection mode. This mode detects the beeps played at the end of an answering machine message, by using the X-Detect header extension. The device sends a SIP INFO message containing the field values Type=AMD and SubType=Beep. This feature allows users of certain third-party, Application server to leave a voice message after an answering machine plays the “beep”. ▪ **[0]** Disabled (default) ▪ **[1]** Start After AMD ▪ **[2]** Start Immediately
Web: Answer Machine Detector Beep Detection Timeout EMS: Beep Detection Timeout **[AMDBeepDetectionTimeout]**	Determines the AMD beep detection timeout (i.e., the duration that the beep detector functions from when detection is initiated). This is used for detecting beeps at the end of an answering machine message. The valid value is in units of 100 milliseconds, from 0 to 1638. The default value is 200 (i.e., 20 seconds).
Web: Answer Machine Detector Beep Detection Sensitivity EMS: Beep Detection Sensitivity **[AMDBeepDetectionSensitivity]**	Determines the AMD beep detection sensitivity for detecting beeps at the end of an answering machine message. The valid value is 0 to 3, where 0 (default) is the least sensitive.

Parameter	Description
Energy Detector Parameters	
Enable Energy Detector **[EnableEnergyDetector]**	Currently, not supported.
Energy Detector Quality Factor **[EnergyDetectorQualityFactor]**	Currently, not supported.
Energy Detector Threshold **[EnergyDetectorThreshold]**	Currently, not supported.
Pattern Detection Parameters **Note:** For an overview on the pattern detector feature for TDM tunneling, refer to DSP Pattern Detector on page 638.	
Web: Enable Pattern Detector **[EnablePatternDetector]**	Enables or disables the activation of the Pattern Detector (PD). Valid options include: ▪ **[0]** Disable = Disable (default) ▪ **[1]** Enable = Enable
[PDPattern]	Defines the patterns that can be detected by the Pattern Detector. The valid range is 0 to 0xFF. **Note:** For this parameter to take effect, a device reset is required.
[PDThreshold]	Defines the number of consecutive patterns to trigger the pattern detection event. The valid range is 0 to 31. The default is 5. **Note:** For this parameter to take effect, a device reset is required.
VXML Parameters	
Web/EMS: Enable VXML **[EnableVXML]**	Enables the VXML stack. ▪ **[0]** Disable = disables VXML stack (default) ▪ **[1]** Enable = enables VXML **Note:** For this parameter to take effect, a device reset is required.
Web: VXML ID **[VXMLID]**	VoiceXML identification string (up to 16 characters) for identifying an incoming VXML call. The default value is 'dialog'.
[VxmlBargeInAllowed]	VXML property that indicates if prompts can be interrupted. ▪ **[0]** = prompts cannot be interrupted ▪ **[1]** = prompts can be interrupted (default) **Note:** For this parameter to take effect, a device reset is required.
[VxmlBuiltinGrammarPath]	Defines the path on the remote Automatic Speech Recognition (ASR) / text-to-speech (TTS) server to access the built-in grammars. The path must not end in a forward slash ('/') as this is added as needed during runtime. The default value is NULL. **Note:** For this parameter to take effect, a device reset is required.

Parameter	Description
[VxmlCompleteTimeout]	Optional parameter that defines the amount of silence (in msec) to wait after speech grammar has been matched before reporting the match. The default value is 0 (i.e., don't set this parameter on recognition attempt). **Note:** For this parameter to take effect, a device reset is required.
[VxmlConfidenceLevel]	Defines the default speech recognition confidence threshold for VXML. The range is from 0 to 100. The default value is 50. **Note:** For this parameter to take effect, a device reset is required.
[VxmlDefaultLanguage]	Defines the default language for speech recognition, if speech recognition has been enabled. If the root document doesn't specify a language and a field or menu element generates speech recognition requests using the GRXML MIME type, the default language is used in the request. The default value is 'en_us'. **Note:** For this parameter to take effect, a device reset is required.
[VxmlIncompleteTimeout]	Optional parameter that defines the amount of silence (in msec) to wait after speech grammar has not matched a voice grammar. The default value is 0 (i.e., don't set this parameter on recognition attempt). **Note:** For this parameter to take effect, a device reset is required.
[VxmlInterDigitTimeout]	Defines the inter-digit timeout value (in msec) used when DTMF is received. The valid range for this parameter is 0 to 7,000 msec. The default value is 3,000. **Note:** For this parameter to take effect, a device reset is required.
[VxmlMaxActiveFiles]	Indicates the maximum number of static VXML scripts that can be loaded to the system at any one time. The valid range for this parameter is 0 to 30. The default value is 10. **Note:** This parameter does not affect the number of dynamic scripts that can be simultaneously active. **Note:** For this parameter to take effect, a device reset is required.

Parameter	Description
[VxmlMaxPorts]	Defines the number of channels in the system that can simultaneously run VXML scripts. The range is from 0 to the maximum number of channels in the system. This value can be used to ensure there are sufficient VXML resources for each call. For example, if the system is running dynamic scripts that each requires many resources, the VxmlMaxPorts value can be lowered to help ensure that each individual call has adequate resources. The default value is 0. **Note:** For this parameter to take effect, a device reset is required.
[VxmlMaxSpeechTimeout]	Defines the maximum time the caller can speak (in msec) in an attempt to match a speech grammar before a no match event is thrown. The range is 0 - 7,000. The default value is 0 (i.e., no time limit in the speech recognition attempt). **Note:** For this parameter to take effect, a device reset is required.
[VxmlNoInputTimeout]	Defines the no input timeout for digit (DTMF) collection or speech recognition (in msec). The range is 0 - 7,000. The default value is 3,000. **Note:** For this parameter to take effect, a device reset is required.
[VxmlSensitivityLevel]	Defines the default speech recognition sensitivity level for VXML. The valid range for this parameter is 0 to 100. The default value is 50. **Note:** For this parameter to take effect, a device reset is required.
[VxmlSpeedVsAccuracy]	Hint to the speech recognition engine for the balance of speed vs. accuracy. The valid range is from 0 to 100. The default value is 50. A low number means the speech recognition engine must perform recognition rapidly, at the cost (i.e., trade off) of accuracy. A high number, such as 100, means the speech recognition engine must perform the speech recognition accurately, at the cost of speed. **Note:** For this parameter to take effect, a device reset is required.
[VxmlSystemInputModes]	Indicates which inputs are valid for grammars. ▪ **[0]** = DTMF is valid (default) ▪ **[1]** = Voice is valid ▪ **[2]** = Both are valid **Note:** For this parameter to take effect, a device reset is required.

Parameter	Description
[VxmlTermChar]	Defines the default terminating digit for received DTMF. The default value is 35 (equivalent to ASCII '#'). **Note:** For this parameter to take effect, a device reset is required.
[VxmlTermTimeout]	Defines the time to wait before terminating received DTMF (in msec). The range is 0 - 7,000. The default value is 3,000. **Note:** For this parameter to take effect, a device reset is required.
Media Resource Control Protocol (MRCP) / Real Time Streaming Protocol (RTSP) Parameters	
[MRCPDefaultMIMEType]	Determines the default format for speech recognition for inline grammars. ▪ **[0]** = indicates GRXML (default) ▪ **[1]** = indicates GL (Nuance format)
[MRCPEnabled]	Activates the Media Resource Control Protocol (MRCP) functionality. ▪ **[0]** = Disable (default) ▪ **[1]** = Activate
[MRCPMaxPorts]	Defines the number of ports that are allocated to running MRCP-related activities such as speech recognition and text-to-speech. A port is considered duplex, so that speech recognition and text-to-speech can run on the same port. The value should not exceed the number of channels in the system. The range is 0 - 120. The default value is 10.
[MRCPServerName]	Defines the hostname of the MRCP server. This is used to build a URI for the server. The default value is NULL.
[MRCPServerIp]	Defines the IP address of the MRCP speech server. The default value is 0.0.0.0.
[MRCPServerPort]	Defines the control port on the MRCP speech server. The range is 0 - 65,535. The default value is 554.
[RTSPConnectionRetryInterval]	Defines the time (in seconds) that the system must wait before trying to create a socket for the RTSP speech server if the socket was never created or was created and then brought down. The range is 0 - 65,535. The default value is 10. **Note:** For this parameter to take effect, a device reset is required.
[RTSPEnabled]	Activates the RTSP functionality. ▪ **[0]** = Disable (default) ▪ **[1]** = Activate **Note:** For this parameter to take effect, a device reset is required.
[RTSPMaxPorts]	Defines the number of channels that can be simultaneously

Parameter	Description
	active in RTSP sessions. The range is 0 - 20.The default value is 10. **Note:** For this parameter to take effect, a device reset is required.

6.11 PSTN Parameters

This subsection describes the device's PSTN parameters.

6.11.1 General Parameters

The general PSTN parameters are described in the table below.

Table 6-49: General PSTN Parameters

Parameter	Description
Web/EMS: Protocol Type **[ProtocolType]**	Defines the PSTN protocol for a the Trunks. To configure the protocol type for a specific Trunk, use the *ini* file parameter ProtocolType_x: ▪ **[0]** NONE ▪ **[1]** E1 EURO ISDN = ISDN PRI Pan-European (CTR4) protocol ▪ **[2]** T1 CAS = Common T1 robbed bits protocols including E&M wink start, E&M immediate start, E&M delay dial/start and loop-start and ground start. ▪ **[3]** T1 RAW CAS ▪ **[4]** T1 TRANSPARENT = Transparent protocol, where no signaling is provided by the device. Timeslots 1 to 24 of all trunks are mapped to DSP channels. ▪ **[5]** E1 TRANSPARENT 31 = Transparent protocol, where no signaling is provided by the device. Timeslots 1 to 31 of each trunk are mapped to DSP channels. ▪ **[6]** E1 TRANSPARENT 30 = Transparent protocol, where no signaling is provided by the device. Timeslots 1 to 31, excluding time slot 16 of all trunks are mapped to DSP channels. ▪ **[7]** E1 MFCR2 = Common E1 MFC/R2 CAS protocols (including line signaling and compelled register signaling). ▪ **[8]** E1 CAS = Common E1 CAS protocols (including line signaling and MF/DTMF address transfer). ▪ **[9]** E1 RAW CAS ▪ **[10]** T1 NI2 ISDN = National ISDN 2 PRI protocol ▪ **[11]** T1 4ESS ISDN = ISDN PRI protocol for the Lucent™/AT&T™ 4ESS switch. ▪ **[12]** T1 5ESS 9 ISDN = ISDN PRI protocol for the Lucent™/AT&T™ 5ESS-9 switch. ▪ **[13]** T1 5ESS 10 ISDN = ISDN PRI protocol for the Lucent™/AT&T™ 5ESS-10 switch.

Parameter	Description
	▪ **[14]** T1 DMS100 ISDN = ISDN PRI protocol for the Nortel™ DMS switch. ▪ **[15]** J1 TRANSPARENT ▪ **[16]** T1 NTT ISDN = ISDN PRI protocol for the Japan - Nippon Telegraph Telephone (known also as INS 1500). ▪ **[17]** E1 AUSTEL ISDN = ISDN PRI protocol for the Australian Telecom. ▪ **[18]** T1 HKT ISDN = ISDN PRI protocol for the Hong Kong - HKT. ▪ **[19]** E1 KOR ISDN = ISDN PRI protocol for Korean Operator (similar to ETSI). ▪ **[20]** T1 HKT ISDN = ISDN PRI protocol for the Hong Kong - HKT. ▪ **[21]** E1 QSIG = ECMA 143 QSIG over E1 ▪ **[22]** E1 TNZ = ISDN PRI protocol for Telecom New Zealand (similar to ETSI) ▪ **[23]** T1 QSIG = ECMA 143 QSIG over T1 ▪ **[30]** E1 FRENCH VN6 ISDN = France Telecom VN6 ▪ **[31]** E1 FRENCH VN3 ISDN = France Telecom VN3 ▪ **[32]** T1 EURO ISDN =ISDN PRI protocol for Euro over T1 ▪ **[35]** T1 DMS100 Meridian ISDN = ISDN PRI protocol for the Nortel™ DMS Meridian switch ▪ **[36]** T1 NI1 ISDN = National ISDN 1 PRI protocol ▪ **[40]** E1 NI2 ISDN = National ISDN 2 PRI protocol over E1 ▪ **[41]** E1 CAS R15 ▪ **[50]** BRI EURO ISDN = Euro ISDN over BRI ▪ **[54]** BRI QSIG = QSIG over BRI ▪ **[55]** BRI FRENCH VN6 ISDN = VN6 over BRI ▪ **[56]** BRI NTT = BRI ISDN Japan (Nippon Telegraph) **Notes:** ▪ All PRI trunks must be configured as the same line type (either E1 or T1). The device can support different variants of CAS and PRI protocols on different E1/T1 spans (no more than four simultaneous PRI variants). ▪ BRI trunks can operate with E1 or T1 trunks.
[ProtocolType_x]	Same as the description for the parameter ProtocolType, but for a specific trunk ID (where x denotes the Trunk ID and 0 is the first trunk).
[ISDNTimerT310]	Defines the T310 override timer for DMS and Euro ISDN variants. An ISDN timer is started when a Q.931 Call Proceeding message is received. The timer is stopped when a Q.931 Alerting/Connect/Disconnect message is received from the other end. The call clears on expiration of the T310 timer. The valid value range is 0 to 600. The default is 0 (i.e., 10 seconds). **Notes:** ▪ For this parameter to take effect, a device reset is required.

Parameter	Description
	▪ When both the parameters ISDNDmsTimerT310 and ISDNTimerT310 are configured, the value of the parameter ISDNTimerT310 prevails.
[ISDNDMSTimerT310]	Overrides the T310 timer for the DMS-100 ISDN variant. T310 defines the timeout between the receipt of a Proceeding message and the receipt of an Alerting/Connect message. The valid range is 10 to 30. The default value is 10 (seconds). **Notes:** ▪ Instead of configuring this parameter, it is recommended to use the parameter ISDNTimerT310. ▪ This parameter is applicable only to Nortel DMS and Nortel MERIDIAN PRI variants (ProtocolType = 14 and 35).
[ISDNJapanNTTTimerT3JA]	T3_JA timer (in seconds). This parameter overrides the internal PSTN T301 timeout on the Users Side (TE side). If an outgoing call from the device to ISDN is not answered during this timeout, the call is released. The valid range is 10 to 240. The default value is 50. **Notes:** ▪ This timer is also affected by the parameter PSTNAlertTimeout. ▪ This parameter is applicable only to the Japan NTT PRI variant (ProtocolType = 16).
Web/EMS: Trace Level **[TraceLevel]**	Defines the trace level: ▪ **[0]** No Trace (default) ▪ **[1]** Full ISDN Trace ▪ **[2]** Layer 3 ISDN Trace ▪ **[3]** Only ISDN Q.931 Messages Trace ▪ **[4]** Layer 3 ISDN No Duplication Trace
Web/EMS: Framing Method **[FramingMethod]**	Determines the physical framing method for the trunk. ▪ **[0]** Extended Super Frame = (Default) Depends on protocol type: ✓ E1: E1 CRC4 MultiFrame Format extended G.706B (same as c) ✓ T1: T1 Extended Super Frame with CRC6 (same as D) ▪ **[1]** Super Frame = T1 SuperFrame Format (as B). ▪ **[a]** E1 FRAMING DDF = E1 DoubleFrame Format - CRC4 is forced to off ▪ **[b]** E1 FRAMING MFF CRC4 = E1 CRC4 MultiFrame Format - CRC4 is always on ▪ **[c]** E1 FRAMING MFF CRC4 EXT = E1 CRC4 MultiFrame Format extended G.706B - auto negotiation is on. If the negotiation fails, it changes automatically to CRC4 off (ddf) ▪ **[A]** T1 FRAMING F4 = T1 4-Frame multiframe. ▪ **[B]** T1 FRAMING F12 = T1 12-Frame multiframe (D4). ▪ **[C]** T1 FRAMING ESF = T1 Extended SuperFrame without CRC6 ▪ **[D]** T1 FRAMING ESF CRC6 = T1 Extended SuperFrame with CRC6

Parameter	Description
	▪ **[E]** T1 FRAMING F72 = T1 72-Frame multiframe (SLC96) ▪ **[F]** T1 FRAMING ESF CRC6 J2 = J1 Extended SuperFrame with CRC6 (Japan) **Note:** This parameter is not configurable for BRI interfaces; the device automatically uses the BRI framing method.
[FramingMethod_x]	Same as the description for parameter FramingMethod, but for a specific trunk ID (where *x* denotes the Trunk ID and 0 is the first Trunk).
Web/EMS: Clock Master **[ClockMaster]**	Determines the Tx clock source of the E1/T1 line. ▪ **[0]** Recovered = Generate the clock according to the Rx of the E1/T1 line (default). ▪ **[1]** Generated = Generate the clock according to the internal TDM bus. **Notes:** ▪ The source of the internal TDM bus clock is determined by the parameter TDMBusClockSource. ▪ For detailed information on configuring the device's clock settings, refer to "Clock Settings" on page 625.
[ClockMaster_x]	Same as the description for parameter ClockMaster, but for a specific Trunk ID (where *x* denotes the Trunk ID and 0 is the first Trunk).
Web/EMS: Line Code **[LineCode]**	Selects B8ZS or AMI for T1 spans, and HDB3 or AMI for E1 spans. ▪ **[0]** B8ZS = use B8ZS line code (for T1 trunks only) default. ▪ **[1]** AMI = use AMI line code. ▪ **[2]** HDB3 = use HDB3 line code (for E1 trunks only). **Note:** This parameter is not configurable for BRI interfaces; the device automatically uses the Modified Alternate Mark Invert (MAMI) line code.
[LineCode_x]	Same as the description for parameter LineCode, but for a specific trunk ID (where 0 depicts the first trunk).
[TrunkLifeLineType]	Defines the type of trunk Lifeline. Short trunks 1-2, 3-4. ▪ **[0]** = Activate Lifeline on power down (default). ▪ **[1]** = Activate Lifeline on power down or on detection of LAN disconnection. ▪ **[2]** = Activate Lifeline on power down or on detection of LAN disconnection or loss of ping. **Note:** For this parameter to take effect, a device reset is required.
[TrunkAdministrativeState]	Defines the administrative state of a trunk. ▪ **[0]** = Lock the trunk; stops trunk traffic to configure the trunk protocol type. ▪ **[2]** = Unlock the trunk (default); enables trunk traffic.
Web/EMS: Line Build Out Loss **[LineBuildOut.**Loss]	Defines the line build out loss for the selected T1 trunk. ▪ **[0]** 0 dB (default) ▪ **[1]** -7.5 dB ▪ **[2]** -15 dB

Parameter	Description
	▪ **[3]** -22.5 dB **Note:** This parameter is applicable only to T1 trunks.
[TDMHairPinning]	Defines static TDM hair-pinning (cross-connection) performed at initialization. The connection is between trunks with an option to exclude a single B-Channel in each trunk. Format example: T0-T1/B3,T2-T3,T4-T5/B2. **Note:** For this parameter to take effect, a device reset is required.
Web: Enable TDM Tunneling EMS: TDM Over IP **[EnableTDMoverIP]**	Enables TDM tunneling. ▪ **[0]** Disable = Disabled (default). ▪ **[1]** Enable = TDM Tunneling is enabled. When TDM Tunneling is enabled, the originating device automatically initiates SIP calls from all enabled B-channels pertaining to E1/T1/J1 spans that are configured with the 'Transparent' protocol. The called number of each call is the internal phone number of the B-channel from where the call originates. The 'Inbound IP Routing Table is used to define the destination IP address of the terminating device. The terminating device automatically answers these calls if its E1/T1 protocol is set to 'Transparent' (ProtocolType = 5). **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ For an overview on TDM tunneling, refer to TDM Tunneling on page 635.

6.11.2 TDM Bus and Clock Timing Parameters

The TDM Bus parameters are described in the table below.

Table 6-50: TDM Bus and Clock Timing Parameters

Parameter	Description
TDM Bus Parameters	
Web/EMS: PCM Law Select **[PCMLawSelect]**	Determines the type of PCM companding law in input/output TDM bus. ▪ **[1]** Alaw = Alaw (default) ▪ **[3]** MuLaw = MuLaw **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ Typically, A-Law is used for E1 spans and Mu-Law for T1/J1 spans.
Web/EMS: Idle PCM Pattern **[IdlePCMPattern]**	Defines the PCM Pattern that is applied to the E1/T1 timeslot (B-channel) when the channel is idle. The range is 0 to 255. The default is set internally according to the Law select 1 (0xFF for Mu-Law; 0x55 for A-law). **Note:** For this parameter to take effect, a device reset is required.

Parameter	Description
Web/EMS: Idle ABCD Pattern **[IdleABCDPattern]**	Defines the ABCD (CAS) Pattern that is applied to the CAS signaling bus when the channel is idle. The valid range is 0x0 to 0xF. The default is -1 (i.e., default pattern is 0000). **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ This parameter is applicable only when using PSTN interface with CAS protocols.
Web/EMS: TDM Bus Clock Source **[TDMBusClockSource]**	Selects the clock source to which the device synchronizes. ▪ **[1]** Internal = Generate clock from local source (default). ▪ **[4]** Network = Recover clock from PSTN line. For detailed information on configuring the device's clock settings, refer to "Clock Settings" on page 625.
EMS/Web: TDM Bus Local Reference **[TDMBusLocalReference]**	Physical Trunk ID from which the device recovers (receives) its clock synchronization. The range is 0 to the maximum number of Trunks. The default is 0. **Note:** This parameter is applicable only if the parameter TDMBusClockSource is set to 4 and the parameter TDMBusPSTNAutoClockEnable is set to 0.
Web/EMS: TDM Bus Enable Fallback **[TDMBusEnableFallback]**	Defines the automatic fallback of the clock. ▪ **[0]** Manual (default) ▪ **[1]** Auto Non-Revertive ▪ **[2]** Auto Revertive
Web: TDM Bus Fallback Clock Source EMS: TDM Bus Fallback Clock **[TDMBusFallbackClock]**	Selects the fallback clock source on which the device synchronizes in the event of a clock failure. ▪ **[4]** Network (default) ▪ **[8]** H.110_A ▪ **[9]** H.110_B ▪ **[10]** NetReference1 ▪ **[11]** NetReference2
Web/EMS: TDM Bus Net Reference Speed **[TDMBusNetrefSpeed]**	Determines the NetRef frequency (for both generation and synchronization). ▪ **[0]** 8 kHz (default) ▪ **[1]** 1.544 MHz ▪ **[2]** 2.048 MHz

Parameter	Description
Web: TDM Bus PSTN Auto FallBack Clock EMS: TDM Bus Auto Fall Back Enable **[TDMBusPSTNAutoClockEnable]**	Enables or disables the PSTN trunk Auto-Fallback Clock feature. ▪ **[0]** Disable (default) = Recovers the clock from the E1/T1 line defined by the parameter TDMBusLocalReference. ▪ **[1]** Enable = Recovers the clock from any connected synchronized slave E1/T1 line. If this trunk loses its synchronization, the device attempts to recover the clock from the next trunk. Note that initially, the device attempts to recover the clock from the trunk defined by the parameter TDMBusLocalReference. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ This parameter is relevant only if the parameter TDMBusClockSource is set to 4.
Web: TDM Bus PSTN Auto Clock Reverting EMS: TDM Bus Auto Fall Back Reverting Enable **[TDMBusPSTNAutoClockRevertingEnable]**	Enables or disables the PSTN trunk Auto-Fallback Reverting feature. If enabled and a trunk returning to service has an AutoClockTrunkPriority parameter value that is higher than the priority of the local reference trunk (set in the TDMBusLocalReference parameter), the local reference reverts to the trunk with the higher priority that has returned to service for the device's clock source. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ This parameter is applicable only when the TDMBusPSTNAutoClockEnable parameter is set to 1.
Web: Auto Clock Trunk Priority EMS: Auto Trunk Priority **[AutoClockTrunkPriority]**	Defines the trunk priority for auto-clock fallback (per trunk parameter). ▪ 0 to 99 = priority, where 0 (default) is the highest. ▪ 100 = the SW never performs a fallback to that trunk (usually used to mark untrusted source of clock). **Note:** Fallback is enabled when the TDMBusPSTNAutoClockEnable parameter is set to 1.

6.11.3 CAS Parameters

The Common Channel Associated (CAS) parameters are described in the table below. Note that CAS is not applicable to BRI interfaces.

Table 6-51: CAS Parameters

Parameter	Description
Web: CAS Transport Type EMS: CAS Relay Transport Mode **[CASTransportType]**	Controls the ABCD signaling transport type over IP. ▪ **[0]** CAS Events Only = Disable CAS relay (default). ▪ **[1]** CAS RFC2833 Relay = Enable CAS relay mode using RFC 2833. The CAS relay mode can be used with the TDM tunneling feature to enable tunneling over IP for both voice and CAS signaling bearers.
[CASAddressingDelimiters]	Determines if delimiters are added to the received address or received ANI digits string. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable When this parameter is enabled, delimiters such as '*', '#', and 'ST' are added to the received address or received ANI digits string. When it is disabled, the address and ANI strings remain without delimiters.
[CASDelimitersPaddingUsage]	Defines the digits string delimiter padding usage per trunk. ▪ **[0]** (default) = default address string padding: '*XXX#' (where XXX is the digit string that begins with '*' and ends with '#', when using padding). ▪ **[1]** = special use of asterisks delimiters: '*XXX*YYY*' (where XXX is the address, YYY is the source phone number, and '*' is the only delimiter padding). **Note:** For this parameter to take effect, a device reset is required.

Web: CAS Table per Trunk
EMS: Trunk CAS Table Index
[CASTableIndex_x]
	Defines the CAS protocol per trunk (where *x* denotes the trunk ID) from a list of CAS protocols defined by the parameter CASFileName_x.
	For example, the below configuration specifies Trunks 0 and 1 to use the E&M Winkstart CAS (E_M_WinkTable.dat) protocol, and Trunks 2 and 3 to use the E&M Immediate Start CAS (E_M_ImmediateTable.dat) protocol:
	`CASFileName_0 = 'E_M_WinkTable.dat'`
	`CASFileName_1 = 'E_M_ImmediateTable.dat'`
	`CASTableIndex_0 = 0`
	`CASTableIndex_1 = 0`
	`CASTableIndex_2 = 1`
	`CASTableIndex_3 = 1`
	Note: You can define CAS tables per B-channel using the parameter CASChannelIndex.

Parameter	Description

Web: Dial Plan
EMS: Dial Plan Name
[CASTrunkDialPlanName_x]
	The CAS Dial Plan name that is used on a specific trunk (where *x* denotes the trunk ID).
	The range is up to 11 characters.
	For example, the below configures E1_MFCR2 trunk with a single protocol (Trunk 5):
	`ProtocolType_5 = 7`
	`CASFileName_0='R2_Korea_CP_ANI.dat'`
	`CASTableIndex_5 = 0`
	`DialPlanFileName = 'DialPlan_USA.dat'`
	`CASTrunkDialPlanName_5 = 'AT_T'`

[CASFileName_x]	CAS file name (e.g., 'E_M_WinkTable.dat') that defines the CAS protocol, where x denotes the CAS file ID (0-7). It is possible to define up to eight different CAS files by repeating this parameter. Each CAS file can be associated with one or more of the device's trunks, using the parameter CASTableIndex_x. **Note:** For this parameter to take effect, a device reset is required.

Web: CAS Table per Channel
[CASChannelIndex]
	Defines the loaded CAS protocol table index per B-channel pertaining to a CAS trunk. This parameter is assigned a string value and can be set in one of the following two formats:
	▪ **CAS table per channel:** Each channel is separated by a comma and the value entered depicts the CAS table index used for that channel. The syntax is <CAS index>,<CAS index> (e.g., "1,2,1,2…"). For this format, 31 indices must be defined for E1 trunks (including dummy for B-channel 16), or 24 indices for T1 trunks. Below is an example for configuring a T1 CAS trunk (Trunk 5) with several CAS variants
	`ProtocolType_5 = 7`
	`CASFILENAME_0='E_M_FGBWinkTable.dat'`
	`CASFILENAME_1='E_M_FGDWinkTable.dat'`
	`CASFILENAME_2='E_M_WinkTable.txt'`
	`CasChannelIndex_5 =`
	`'0,0,0,1,1,1,2,2,2,0,0,0,1,1,1,0,1,2,0,2,1,2,`
	`2,2'`
	`CASDelimitersPaddingUsage_5 = 1`
	▪ **CAS table per channel group:** Each channel group is separated by a colon and each channel is separated by a comma. The syntax is <x-y channel range>:<CAS table index>, (e.g., "1-10:1,11-31:3"). Every B-channel (including 16 for E1) must belong to a channel group. Below is an example for configuring an E1 CAS trunk (Trunk 5) with several CAS variants:
	`ProtocolType_5 = 8`
	`CASFILENAME_2='E1_R2D'`
	`CASFILENAME_7= E_M_ImmediateTable_A-Bit.txt'`
	`CasChannelIndex_5 = '1-10:2,11-20:7,21-31:2'`
	Notes:
	▪ Only one of these formats can be implemented; not both.
	▪ When this parameter is not configured, a single CAS table for the entire trunk is used, configured by the parameter CASTableIndex.

Parameter	Description
[CASTablesNum]	Indicates how many CAS protocol configurations files are loaded. The valid range is 1 to 8. **Note:** For this parameter to take effect, a device reset is required.
CAS State Machines Parameters **Note:** For configuring the 'CAS State Machine' table using the Web interface, refer to "Configuring the CAS State Machines" on page 77.	
Web: Generate Digit On Time **[CASStateMachineGenerateDigitOnTime]**	Generates digit on-time (in msec). The value must be a positive value. The default value is -1.
Web: Generate Inter Digit Time **[CASStateMachineGenerateInterDigitTime]**	Generates digit off-time (in msec). The value must be a positive value. The default value is -1.
Web: DTMF Max Detection Time **[CASStateMachineDTMFMaxOnDetectionTime]**	Detects digit maximum on time (according to DSP detection information event) in msec units. The value must be a positive value. The default value is -1.
Web: DTMF Min Detection Time **[CASStateMachineDTMFMinOnDetectionTime]**	Detects digit minimum on time (according to DSP detection information event) in msec units. The digit time length must be longer than this value to receive a detection. Any number may be used, but the value must be less than CasStateMachineDTMFMaxOnDetectionTime. The value must be a positive value. The default value is -1.
Web: MAX Incoming Address Digits **[CASStateMachineMaxNumOfIncomingAddressDigits]**	Defines the limitation for the maximum address digits that need to be collected. After reaching this number of digits, the collection of address digits is stopped. The value must be an integer. The default value is -1.
Web: MAX Incoming ANI Digits **[CASStateMachineMaxNumOfIncomingANIDigits]**	Defines the limitation for the maximum ANI digits that need to be collected. After reaching this number of digits, the collection of ANI digits is stopped. The value must be an integer. The default value is -1.
Web: Collect ANI **[CASStateMachineCollectANI]**	In some cases, when the state machine handles the ANI collection (not related to MFCR2), you can control the state machine to collect ANI or discard ANI. ▪ **[0]** No = Don't collect ANI. ▪ **[1]** Yes = Collect ANI. ▪ **[-1]** Default = Default value.
Web: Digit Signaling System **[CASStateMachineDigitSignalingSystem]**	Defines which Signaling System to use in both directions (detection\generation). ▪ **[0]** DTMF = Uses DTMF signaling. ▪ **[1]** MF = Uses MF signaling (default). ▪ **[-1]** Default = Default value.

6.11.4 ISDN Parameters

The ISDN parameters are described in the table below.

Table 6-52: ISDN Parameters

Parameter	Description
Web: ISDN Termination Side EMS: Termination Side **[TerminationSide]**	Selects the ISDN termination side. ▪ **[0]** User side = ISDN User Termination Equipment (TE) side (default) ▪ **[1]** Network side = ISDN Network Termination (NT) side **Note:** Select 'User side' when the PSTN or PBX side is configured as 'Network side' and vice versa. If you don't know the device's ISDN termination side, choose 'User side'. If the D-channel alarm is indicated, choose 'Network Side'. The BRI module supports the ITU-T I.430 standard, which defines the ISDN-BRI layer 1 specification. The BRI and PRI ports are configured similarly, using this parameter. When an NT port is active, it drives a 38-V line and sends an INFO1 signal (as defined in ITU-T I.430 Table 4) on the data line to synchronize to a TE port that might be connected to it. To stop the voltage and the INFO1 signal on the line, stop the trunk using the Stop Trunk button.
[TerminationSide_x]	Same as the description for parameter TerminationSide, but for a specific trunk ID (where *x* denotes the Trunk ID and 0 is the first Trunk).
BRI Layer 2 Mode **[BriLayer2Mode]**	Determines whether Point-to-Point or Point-to-Multipoint mode for BRI ports. ▪ **[0]** Point to Point (default) ▪ **[1]** Point to Multipoint = Must be configured for Network side.
Web/EMS: B-channel Negotiation **[BchannelNegotiation]**	Determines the ISDN B-Channel negotiation mode. ▪ **[0]** Preferred. ▪ **[1]** Exclusive (default). ▪ **[2]** Any. **Notes:** ▪ This parameter is applicable only to ISDN protocols. ▪ For some ISDN variants, when 'Any' (2) is selected, the Setup message excludes the Channel Identification IE. ▪ The 'Any' (2) option is applicable only if the following conditions are met: ✓ The parameter TerminationSide is set to 0 ('User side'). ✓ The PSTN protocol type (ProtocolType) is configured as Euro ISDN.

Parameter	Description
NFAS Parameters **Note:** These parameters are not applicable to BRI interfaces.	
Web: NFAS Group Number EMS: Group Number **[NFASGroupNumber_x]**	Indicates the NFAS group number (NFAS member) for the selected trunk, where x depicts the Trunk ID. ▪ 0 = Non-NFAS trunk (default) ▪ 1 to 9 = NFAS group number Trunks that belong to the same NFAS group have the same number. With ISDN Non-Facility Associated Signaling you can use single D-channel to control multiple PRI interfaces. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ This parameter is applicable only to T1 ISDN protocols. ▪ For a detailed description on NFAS, refer to ISDN Non-Facility Associated Signaling (NFAS) on page 631.
Web/EMS: D-channel Configuration **[DChConfig_x]**	Defines primary, backup (optional), and B-channels only, per trunk (where x depicts the Trunk ID). ▪ **[0]** PRIMARY= Primary Trunk (default) - contains a D-channel that is used for signaling. ▪ **[1]** BACKUP = Backup Trunk - contains a backup D-channel that is used if the primary D-channel fails. ▪ **[2]** NFAS = NFAS Trunk - contains only 24 B-channels, without a signaling D-channel. **Note:** This parameter is applicable only to T1 ISDN protocols.
Web: NFAS Interface ID EMS: ISDN NFAS Interface ID **[ISDNNFASInterfaceID_x]**	Defines a different Interface ID for each T1 trunk (where x denotes the trunk ID). The valid range is 0 to 100. The default interface ID equals the trunk's ID. **Notes:** ▪ To set the NFAS interface ID, configure ISDNIBehavior_x to include '512' feature per T1 trunk. ▪ For a detailed description on NFAS, refer to ISDN Non-Facility Associated Signaling (NFAS) on page 631.
Web: Enable ignoring ISDN Disconnect with PI **[KeepISDNCallOnDisconnectWithPI]**	Allows the device to ignore ISDN Disconnect messages with PI 1 or 8. ▪ **[1]** = The call (in connected state) is not released if a Q.931 Disconnect with PI (PI = 1 or 8) message is received during the call. ▪ **[0]** = The call is disconnected (default).

Parameter	Description
Web: PI For Setup Message **[PIForSetupMsg]**	Determines whether and which Progress Indicator (PI) information element (IE) is added to the sent ISDN Setup message. Some ISDN protocols such as NI-2 or Euro ISDN can optionally contain PI = 1 or PI = 3 in the Setup message. ▪ **[0]** = PI is not added (default). ▪ **[1]** = PI 1 is added to a sent ISDN Setup message - call is not end-to-end ISDN. ▪ **[3]** = PI 3 is added to a sent ISDN Setup message - calling equipment is not ISDN.
ISDN Flexible Behavior Parameters ISDN protocol is implemented in different switches/PBXs by different vendors. Several implementations may vary slightly from the specification. Therefore, to provide a flexible interface that supports these ISDN variants, the ISDN behavior parameters can be used.	
Web/EMS: Incoming Calls Behavior **[ISDNInCallsBehavior]**	The bit-field used to determine several behavior options that influence how the ISDN Stack INCOMING calls behave. ▪ **[32]** DATA CONN RS = The device sends a Connect (answer) message on not incoming Tel calls. ▪ **[64]** VOICE CONN RS = The device sends a Connect (answer) message on incoming Tel calls. ▪ **[2048]** CHAN ID IN FIRST RS = The device sends Channel ID in the first response to an incoming Q.931 Call Setup message. Otherwise, the Channel ID is sent only if the device requires changing the proposed Channel ID (default). ▪ **[8192]** CHAN ID IN CALL PROC = The device sends Channel ID in a Q.931 Call Proceeding message. ▪ **[65536]** PROGR IND IN SETUP ACK = The device includes Progress Indicator (PI=8) in Setup ACK message if an empty called number is received in an incoming Setup message. This option is applicable to the overlap dialing mode. The device also plays a dial tone (for TimeForDialTone) until the next called number digits are received. ▪ **[262144]** = NI-2 second redirect number. You can select and use (in INVITE messages) the NI-2 second redirect number if two redirect numbers are received in Q.931 Setup for incoming Tel-to-IP calls. ▪ **[2147483648]** CC_USER_SCREEN_INDICATOR = When the device receives two Calling Number IE's in the Setup message, the device by default, uses only one of the numbers according to the following: ✓ Network provided, Network provided - the first calling number is used ✓ Network provided, User provided: the first one is used ✓ User provided, Network provided: the second one is used ✓ User provided, user provided: the first one is used When this bit is configured, the device behaves as follows:

Parameter	Description
	✓ Network provided, Network provided: the first calling number is used ✓ Network provided, User provided: the second one is used ✓ User provided, Network provided: the first one is used ✓ User provided, user provided: the first one is used **Note:** When using the *ini* file to configure the device to support several ISDNInCallsBehavior features, enter a summation of the individual feature values. For example, to support both **[2048]** and **[65536]** features, set ISDNInCallsBehavior = 67584 (i.e., 2048 + 65536).
[ISDNInCallsBehavior_x]	Same as the description for the parameter ISDNInCallsBehavior, but per trunk (i.e., where *x* depicts the Trunk ID).
Web/EMS: Q.931 Layer Response Behavior **[ISDNIBehavior]**	Bit-field used to determine several behavior options that influence the behaviour of the Q.931 protocol. ▪ **[0]** = Disable (default) ▪ **[1]** NO STATUS ON UNKNOWN IE = Q.931 Status message isn't sent if Q.931 received message contains an unknown/unrecognized IE. By default, the Status message is sent. **Note:** This value is applicable only to ISDN variants in which sending of Status message is optional. ▪ **[2]** NO STATUS ON INV OP IE = Q.931 Status message isn't sent if an optional IE with invalid content is received. By default, the Status message is sent. **Note:** This option is applicable only to ISDN variants in which sending of Status message is optional. ▪ **[4]** ACCEPT UNKNOWN FAC IE = Accepts unknown/unrecognized Facility IE. Otherwise, the Q.931 message that contains the unknown Facility IE is rejected (default). **Note:** This option is applicable only to ISDN variants where a complete ASN1 decoding is performed on Facility IE. ▪ **[128]** SEND USER CONNECT ACK = The Connect ACK message is sent in response to received Q.931 Connect; otherwise, the Connect ACK is not sent (default). **Note:** This option is applicable only to Euro ISDN User side outgoing calls. ▪ **[512]** EXPLICIT INTERFACE ID = Enables to configure T1 NFAS Interface ID (refer to the parameter ISDNNFASInterfaceID_x). **Note:** This value is applicable only to 4/5ESS, DMS, NI-2 and HKT variants. ▪ **[2048]** ALWAYS EXPLICIT = Always set the Channel Identification IE to explicit Interface ID, even if the B-channel is on the same trunk as the D-channel. **Note:** This value is applicable only to 4/5ESS, DMS and NI-2 variants. ▪ **[32768]** ACCEPT MU LAW =Mu-Law is also accepted in ETSI.

Parameter	Description
	▪ **[65536]** EXPLICIT PRES SCREENING = The calling party number (octet 3a) is always present even when presentation and screening are at their default. **Note:** This option is applicable only to ETSI, NI-2, and 5ESS. ▪ **[131072]** STATUS INCOMPATIBLE STATE = Clears the call on receipt of Q.931 Status with incompatible state. Otherwise, no action is taken (default). ▪ **[262144]** STATUS ERROR CAUSE = Clear call on receipt of Status according to cause value. ▪ **[524288]** ACCEPT A LAW =A-Law is also accepted in 5ESS. ▪ **[2097152]** RESTART INDICATION = Upon receipt of a Restart message, acEV_PSTN_RESTART_CONFIRM is generated. ▪ **[4194304]** FORCED RESTART = On data link (re)initialization, send RESTART if there is no call. ▪ **[67108864]** NS ACCEPT ANY CAUSE = Accept any Q.850 cause from ISDN. **Note:** This option is applicable only to Euro ISDN. ▪ **[134217728]** NS_BRI_DL_ALWAYS_UP (0x08000000) = By default, the BRI D-channel goes down if there are no active calls. If this option is configured, the BRI D-channel is always up and synchronized. ▪ **[536870912]** Alcatel coding for redirect number and display name is accepted by the device. **Note:** This option is applicable only to QSIG (and relevant for specific Alcatel PBXs such as OXE). ▪ **[1073741824]** QSI ENCODE INTEGER = If this bit is set, INTEGER ASN.1 type is used in operator coding (compliant to new ECMA standards); otherwise, OBJECT IDENTIFIER ASN.1 type is used. **Note:** This option is applicable only to QSIG. ▪ **[2147483648]** 5ESS National Mode For Bch Maintenance = Use the National mode of AT&T 5ESS for B-channel maintenance. **Notes:** ▪ To configure the device to support several ISDNIBehavior features, enter a summation of the individual feature values. For example, to support both **[512]** and **[2048]** features, set the parameter ISDNIBehavior is set to 2560 (i.e., 512 + 2048). ▪ When configuring in the Web interface, to select the options click the arrow button and then for each required option select 1 to enable.
[ISDNIBehavior_x]	Same as the description for parameter ISDNIBehavior, but for a specific trunk ID.

Parameter	Description
Web: General Call Control Behavior EMS: General CC Behavior **[ISDNGeneralCCBehavior]**	Bit-field for determining several general CC behavior options. To select the options, click the arrow button, and then for each required option, select 1 to enable. The default is 0 (i.e., disable). ▪ **[2]** = Data calls with interworking indication use 64 kbps B-channels (physical only). ▪ **[8]** REVERSE CHAN ALLOC ALGO = Channel ID allocation algorithm. ▪ **[16]** = The device clears down the call if it receives a NOTIFY message specifying 'User-Suspended'. A NOTIFY (User-Suspended) message is used by some networks (e.g., in Italy or Denmark) to indicate that the remote user has cleared the call, especially in the case of a long distance voice call. ▪ **[32]** CHAN ID 16 ALLOWED = Applies only to ETSI E1 lines (30B+D). Enables handling the differences between the newer QSIG standard (ETS 300-172) and other ETSI-based standards (ETS 300-102 and ETS 300-403) in the conversion of B-channel ID values into timeslot values: ✓ In 'regular ETSI' standards, the timeslot is identical to the B-channel ID value, and the range for both is 1 to 15 and 17 to 31. The D-channel is identified as channel-id #16 and carried into the timeslot #16. ✓ In newer QSIG standards, the channel-id range is 1 to 30, but the timeslot range is still 1 to 15 and 17 to 31. The D-channel is not identified as channel-id #16, but is still carried into the timeslot #16. When this bit is set, the channel ID #16 is considered as a valid B-channel ID, but timeslot values are converted to reflect the range 1 to 15 and 17 to 31. This is the new QSIG mode of operation. When this bit is not set (default), the channel_id #16 is not allowed, as for all ETSI-like standards. ▪ **[64]** USE T1 PRI = PRI interface type is forced to T1. ▪ **[128]** USE E1 PRI = PRI interface type is forced to E1. ▪ **[256]** START WITH B CHAN OOS = B-channels start in the Out-Of-Service state (OOS). ▪ **[512]** CHAN ALLOC LOWEST = CC allocates B-channels starting from the lowest available B-channel id. ▪ **[1024]** CHAN ALLOC HIGHEST = CC allocates B-channels starting from the highest available B-channel id. ▪ **[16384]** CC_TRANSPARENT_UUI bit: The UUI-protocol implementation of CC is disabled allowing the application to freely send UUI elements in any primitive, regardless of the UUI-protocol requirements (UUI Implicit Service 1). This allows more flexible application control on the UUI. When this bit is not set (default behavior), CC implements the UUI-protocol as specified in the ETS 300-403 standards for Implicit Service 1. ▪ **[65536]** GTD5 TBCT = CC implements the VERIZON-GTD-5 Switch variant of the TBCT Supplementary

Parameter	Description
	Service, as specified in FSD 01-02-40AG Feature Specification Document from Verizon. Otherwise, TBCT is implemented as specified in GR-2865-CORE specification (default behavior). **Note:** When using the *ini* file to configure the device to support several ISDNGeneralCCBehavior features, add the individual feature values. For example, to support both **[16]** and **[32]** features, set ISDNGeneralCCBehavior = 48 (i.e., 16 + 32).
Web/EMS: Outgoing Calls Behavior **[ISDNOutCallsBehavior]**	This parameter determines several behaviour options (bit fields) that influence the behaviour of the ISDN Stack outgoing calls. To select options, click the arrow button, and then for each required option, select 1 to enable. The default is 0 (i.e., disable). ▪ **[2]** USER SENDING COMPLETE =The device doesn't automatically generate the Sending-Complete IE in the Setup message. If this bit is not set, the device generates it automatically in the Setup message only. ▪ **[16]** USE MU LAW = The device sends G.711-m-Law in outgoing voice calls. When disabled, the device sends G.711-A-Law in outgoing voice calls. **Note:** This option is applicable only to the Korean variant. ▪ **[128]** DIAL WITH KEYPAD = The device uses the Keypad IE to store the called number digits instead of the CALLED_NB IE. **Note:** This option is applicable only to the Korean variant (Korean network). This is useful for Korean switches that don't accept the CALLED_NB IE. ▪ **[256]** STORE CHAN ID IN SETUP = The device forces the sending of a Channel-Id IE in an outgoing Setup message even if it's not required by the standard (i.e., optional) and no Channel-Id has been specified in the establishment request. This is useful for improving required compatibility with switches. On BRI lines, the Channel-Id IE indicates 'any channel'. On PRI lines, it indicates an unused channel ID, preferred only. ▪ **[572]** USE A LAW = The device sends G.711 A-Law in outgoing voice calls. When disabled, the device sends the default G.711-Law in outgoing voice calls. **Note:** This option is applicable only to the E10 variant. ▪ **[1024]** = Numbering plan/type for T1 IP-to-Tel calling numbers are defined according to the manipulation tables or according to the RPID header (default). Otherwise, the plan/type for T1 calls are set according to the length of the calling number. ▪ **[2048]** = The device accepts any IA5 character in the called_nb and calling_nb strings and sends any IA5 character in the called_nb, and is not restricted to extended digits only (i.e., 0-9,*,#). ▪ **[16384]** DLCI REVERSED OPTION = Behavior bit used in the IUA interface groups to indicate that the reversed

Parameter	Description
	format of the DLCI field must be used. **Note:** When using the *ini* file to configure the device to support several ISDNOutCallsBehavior features, add the individual feature values. For example, to support both **[2]** and **[16]** features, set ISDNOutCallsBehavior = 18 (i.e., 2 + 16).
[ISDNOutCallsBehavior_x]	Same as the description for parameter ISDNOutCallsBehavior, but for a specific trunk ID.

6.12 ISDN and CAS Interworking Parameters

The ISDN and CAS interworking parameters are described in the table below.

Table 6-53: ISDN and CAS Interworking Parameters

Parameter	Description
ISDN Parameters	
Web/EMS: Min Routing Overlap Digits **[MinOverlapDigitsForRouting]**	Minimum number of overlap digits to collect (for ISDN overlap dialing) before sending the first SIP message for routing Tel-to-IP calls. The valid value range is 0 to 49. The default is 1. **Note:** This parameter is applicable when the ISDNRxOverlap parameter is set to **[2].**
Web/EMS: ISDN Overlap IP to Tel Dialing **[ISDNTxOverlap]**	Enables ISDN overlap dialing for IP-to-Tel calls. This feature is part of ISDN-to-SIP overlap dialing according to RFC 3578. ▪ **[0]** Disable (default) ▪ **[1]** Enable When enabled, for each received INVITE of the same dialog session, the device sends an ISDN Setup (and subsequent ISDN Info Q.931 messages) with the collected digits to the Tel side. For all subsequent INVITEs received, the device sends a SIP 484 Address Incomplete response in order to maintain the current dialog session and receive additional digits from subsequent INVITEs. **Note:** When IP-to-Tel overlap dialing is enabled, to send ISDN Setup message without Sending Complete IE, the CC_USER_SENDING_COMPLETE bit (=2) must be enabled in the ISDNOutgoingCallsBehavior parameter.
Web: Enable Receiving of Overlap Dialing **[ISDNRxOverlap_x]**	Determines the receiving (Rx) type of ISDN overlap dialing for Tel-to-IP calls. ▪ **[0]** None (default) = Disabled. ▪ **[1]** Local receiving = ISDN Overlap Dialing - the complete number is sent in the INVITE Request-URI user part. The device receives ISDN called number that is sent in the 'Overlap' mode. The ISDN Setup message is sent to IP only after the number (including the Sending Complete IE) is fully received (via Setup and/or subsequent Info Q.931 messages). In other words, the device waits until it has

Parameter	Description
	received all the ISDN signaling messages containing parts of the called number, and only then it sends a SIP INVITE with the entire called number in the Request-URI. ▪ **[2]** Through SIP = Interworking of ISDN Overlap Dialing to SIP, based on RFC 3578. The device interworks ISDN to SIP by sending digits each time they are received (from Setup and subsequent Info Q.931 messages) to the IP, using subsequent SIP INVITE messages. **Notes:** ▪ When option **[2]** is configured, you can define the minimum number of overlap digits to collect before sending the first SIP message for routing the call, using the MinOverlapDigitsForRouting parameter. ▪ When option **[2]** is configured, even if SIP 4xx responses are received during this ISDN overlap receiving, the device does not release the call. ▪ The MaxDigits parameter can be used to limit the length of the collected number for ISDN overlap dialing (if Sending Complete is not received). ▪ If a digit map pattern is defined (using the DigitMapping or DialPlanIndex parameters), the device collects digits until a match is found (e.g., for closed numbering schemes) or until a timer expires (e.g., for open numbering schemes). If a match is found (or the timer expires), the digit collection process is terminated even if Sending Complete is not received. ▪ For enabling ISDN overlap dialing for IP-to-Tel calls, use the ISDNTxOverlap parameter. ▪ For detailed information on ISDN overlap dialing, refer to "ISDN Overlap Dialing" on page 630.
[ISDNRxOverlap]	Same as the description for parameter ISDNRxOverlap_x, but for all trunks.
Web/EMS: Mute DTMF In Overlap **[MuteDTMFInOverlap]**	Enables the muting of in-band DTMF detection until the device receives the complete destination number from the ISDN (for Tel-to-IP calls). In other words, the device does not accept DTMF digits received in the voice stream from the PSTN, but only accepts digits from ISDN Info messages. ▪ **[0]** Don't Mute (default) ▪ **[1]** Mute DTMF in Overlap Dialing = The device ignores in-band DTMF digits received during ISDN overlap dialing (disables the DTMF in-band detector). **Notes:** ▪ When enabled and at least one digit is received from the ISDN (Setup message), the device stops playing a dial tone. ▪ This parameter is applicable only to ISDN Overlap mode when dialed numbers are sent using Q.931 Info messages.

Parameter	Description
[ConnectedNumberType]	Defines the Numbering Type of the ISDN Q.931 Connected Number IE that the device sends in the Connect message to the ISDN (for Tel-to-IP calls). This is interworked from the P-Asserted-Identity header in SIP 200 OK. The default is **[0]** (i.e., unknown).
[ConnectedNumberPlan]	Defines the Numbering Plan of the ISDN Q.931 Connected Number IE that the device sends in the Connect message to the ISDN (for Tel-to-IP calls). This is interworked from the P-Asserted-Identity header in SIP 200 OK. The default is **[0]** (i.e., unknown).
Web/EMS: Enable ISDN Tunneling Tel to IP **[EnableISDNTunnelingTel2IP]**	Enables ISDN Tunneling. ▪ **[0]** Disable = Disable (default). ▪ **[1]** Using Header = Enable ISDN Tunneling from ISDN PRI to SIP using a proprietary SIP header. ▪ **[2]** Using Body = Enable ISDN Tunneling from ISDN PRI to SIP using a dedicated message body. When ISDN Tunneling is enabled, the device sends all ISDN PRI messages using the correlated SIP messages. The ISDN Setup message is tunneled using SIP INVITE, all mid-call messages are tunneled using SIP INFO, and ISDN Disconnect/Release message is tunneled using SIP BYE messages. The raw data from the ISDN is inserted into a proprietary SIP header (X-ISDNTunnelingInfo) or a dedicated message body (application/isdn) in the SIP messages. **Note:** For this feature to function, you must set the parameter ISDNDuplicateQ931BuffMode to 128 (i.e., duplicate all messages).
Web/EMS: Enable ISDN Tunneling IP to Tel **[EnableISDNTunnelingIP2Tel]**	Enables ISDN Tunneling to the Tel side. ▪ **[0]** Disable = Disable (default). ▪ **[1]** Using Header = Enable ISDN Tunneling from SIP to ISDN PRI using a proprietary SIP header. ▪ **[2]** Using Body = Enable ISDN Tunneling from SIP to ISDN PRI using a dedicated message body. When ISDN Tunneling is enabled, the device extracts raw data received in a proprietary SIP header (X-ISDNTunnelingInfo) or a dedicated message body (application/isdn) in the SIP messages and sends the data as ISDN messages to the PSTN side.
Web/EMS: Enable QSIG Tunneling **[EnableQSIGTunneling]**	Enables QSIG tunneling-over-SIP according to <draft-elwell-sipping-qsig-tunnel-03>. ▪ **[0]** Disable = Disable (default). ▪ **[1]** Enable = Enable QSIG tunneling from QSIG to SIP and vice versa. When QSIG tunneling is enabled, all QSIG messages are sent as raw data in corresponding SIP messages using a dedicated message body. **Notes:** ▪ QSIG tunneling must be enabled on both originating and

Parameter	Description
	terminating devices. ▪ To enable this function, set the parameter ISDNDuplicateQ931BuffMode to 128 (i.e., duplicate all messages).
Web: Enable Hold to ISDN EMS: Enable Hold 2 ISDN **[EnableHold2ISDN]**	Enables interworking of the Hold/Retrieve supplementary service from SIP to PRI. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Notes:** ▪ This capability is supported only for QSIG and Euro ISDN variants. ▪ For Euro ISDN, it is only supported from TE (user) to NT (network). ▪ If the parameter is disabled (or for other ISDN variants), the device plays a Held tone to the Tel side when a SIP request with 0.0.0.0 or inactive in SDP is received. An appropriate CPT file with the Held tone should be used.
EMS: Duplicate Q931 Buff Mode **[ISDNDuplicateQ931BuffMode]**	Controls the activation/deactivation of delivering raw Q.931 messages. ▪ **[0]** = ISDN messages aren't duplicated (default). ▪ **[128]** = All ISDN messages are duplicated. **Note:** For this parameter to take effect, a device reset is required.
Web/EMS: ISDN SubAddress Format **[ISDNSubAddressFormat]**	Determines the encoding format of the SIP Tel URI parameter 'isub', which carries the encoding type of ISDN subaddresses. This is used to identify different remote ISDN entities under the same phone number - ISDN Calling and Called numbers) for interworking between ISDN and SIP networks. ▪ **[0]** = ASCII - IA5 format that allows up to 20 digits. Indicates that the 'isub' parameter value needs to be encoded using ASCII characters (default) ▪ **[1]** = BCD (Binary Coded Decimal) - allows up to 40 characters (digits and letters). Indicates that the 'isub' parameter value needs to be encoded using BCD when translated to an ISDN message. ▪ **[2]** = User Specified For IP-to-Tel calls, if the incoming SIP INVITE message includes subaddress values in the 'isub' parameter for the Called Number (in the Request-URI) and/or the Calling Number (in the From header), these values are mapped to the outgoing ISDN Setup message. If the incoming ISDN Setup message includes 'subaddress' values for the Called Number and/or the Calling Number, these values are mapped to the outgoing SIP INVITE message's ‘isub’ parameter in accordance with RFC 4715.

Parameter	Description
[IgnoreISDNSubaddress]	Determines whether the device ignores the Subaddress from the incoming ISDN Called and Calling numbers when sending to IP. ▪ **[0]** = If an incoming ISDN Q.931 Setup message contains a Called/Calling Number Subaddress, the Subaddress is interworked to the SIP 'isub' parameter according to RFC (default). ▪ **[1]** = The device removes the ISDN Subaddress and does not include the 'isub' parameter in the Request-URI and does not process INVITEs with this parameter.
Web: Play Busy Tone to Tel **[PlayBusyTone2ISDN]**	Enables the device to play a busy or reorder tone to the PSTN after a Tel-to-IP call is released. ▪ **[0]** Don't Play = Immediately sends an ISDN Disconnect message (default). ▪ **[1]** Play when Disconnecting = Sends an ISDN Disconnect message with PI = 8 and plays a busy or reorder tone to the PSTN (depending on the release cause). ▪ **[2]** Play before Disconnect = Delays the sending of an ISDN Disconnect message for a user-defined time (configured by the TimeForReorderTone parameter) and plays a busy or reorder tone to the PSTN. This is applicable only if the call is released from the IP **[Busy Here** (486) or Not Found (404)] before it reaches the Connect state; otherwise, the Disconnect message is sent immediately and no tones are played.
Web: Play Ringback Tone to Trunk **[PlayRBTone2Trunk_ID]**	Enables the playing of a ringback tone (RBT) to the trunk side and per trunk (where *ID* depicts the trunk number and 0 is the first trunk). This parameter also determines the method for playing the RBT. ▪ **[-1]** = Not configured - use the value of the parameter PlayRBTone2Tel (default). ▪ **[0]** Don't Play = The device configured with ISDN/CAS protocol type does not play an RBT. No PI is sent to the ISDN unless the parameter ProgressIndicator2ISDN_ID is configured differently. ▪ **[1]** Play on Local = The device configured with CAS protocol type plays a local RBT to PSTN upon receipt of a SIP 180 Ringing response (with or without SDP). **Note:** Receipt of a 183 response does not cause the device configured with CAS to play an RBT (unless SIP183Behaviour is set to 1). The device configured with ISDN protocol type operates according to the parameter LocalISDNRBSource: ✓ If the device receives a 180 Ringing response (with or without SDP) and the parameter LocalISDNRBSource is set to 1, it plays an RBT and sends an ISDN Alert with PI = 8 (unless the parameter ProgressIndicator2ISDN_ID is configured differently). ✓ If the parameter LocalISDNRBSource is set to 0, the device doesn't play an RBT and an Alert message (without PI) is sent to the ISDN. In this case, the PBX/PSTN plays the RBT to the originating terminal

Parameter	Description
	by itself. **Note:** Receipt of a 183 response does not cause the device with ISDN protocol type to play an RBT; the device issues a Progress message (unless SIP183Behaviour is set to 1). If the parameter SIP183Behaviour is set to 1, the 183 response is handled the same way as a 180 Ringing response. ▪ **[2]** Prefer IP = Play according to 'Early Media'. If a SIP 180 response is received and the voice channel is already open (due to a previous 183 early media response or due to an SDP in the current 180 response), the device with ISDN/CAS protocol type doesn't play the RBT; PI = 8 is sent in an ISDN Alert message (unless the parameter ProgressIndicator2ISDN_ID is configured differently). If a 180 response is received, but the 'early media' voice channel is not opened, the device with CAS protocol type plays an RBT to the PSTN. The device with ISDN protocol type operates according to the parameter LocalISDNRBSource: ✓ If LocalISDNRBSource is set to 1, the device plays an RBT and sends an ISDN Alert with PI = 8 to the ISDN (unless the parameter ProgressIndicator2ISDN_ID is configured differently). ✓ If LocalISDNRBSource is set to 0, the device doesn't play an RBT. No PI is sent in the ISDN Alert message (unless the parameter ProgressIndicator2ISDN_ID is configured differently). In this case, the PBX/PSTN should play an RBT tone to the originating terminal by itself. **Note:** Receipt of a 183 response results in an ISDN Progress message (unless SIP183Behaviour is set to 1). If SIP183Behaviour is set to 1 (183 is handled the same way as a 180 + SDP), the device sends an Alert message with PI = 8, without playing an RBT. ▪ **[3]** Play tone according to received media. The behaviour is similar to **[2].** If a SIP 180 response is received and the voice channel is already open (due to a previous 183 early media response or due to an SDP in the current 180 response), the device plays a local RBT if there are no prior received RTP packets. The device stops playing the local RBT as soon as it starts receiving RTP packets. At this stage, if the device receives additional 18x responses, it does not resume playing the local RBT. **Note:** For ISDN trunks, this option is applicable only if LocalISDNRBSource is set to 1.
Web: Digital Out-Of-Service Behavior EMS: Digital OOS Behavior For Trunk Value **[DigitalOOSBehaviorFor Trunk_ID]**	Determines the method for setting digital trunks to Out-Of-Service state per trunk. ▪ **[-1]** Not Configured = Use the settings of the DigitalOOSBehavior parameter for per device (default). ▪ **[0]** Default = Uses default behavior for each trunk (see note below). ▪ **[1]** Service = Sends ISDN In or Out of Service (only for ISDN protocols that support Service message). ▪ **[2]** D-Channel = Takes D-Channel down or up (ISDN

Parameter	Description
	only). ▪ **[3]** Alarm = Sends or clears PSTN AIS Alarm (ISDN and CAS). ▪ **[4]** Block = Blocks trunk (CAS only). **Notes:** ▪ This parameter is applicable only if the parameter EnableBusyOut is set to 1. ▪ The default behavior (value 0) is as follows: ✓ **ISDN:** Use Service messages on supporting variants and use Alarm on non-supporting variants. ✓ CAS: Use Alarm. ▪ When updating this parameter value at run-time, you must stop the trunk and then restart it for the update to take effect. ▪ To determine the method for setting Out-Of-Service state for all trunks (i.e., per device), use the DigitalOOSBehavior parameter. ▪ The *ID* in the *ini* file parameter name represents the trunk number, where 0 is the first trunk.
Web: Digital Out-Of-Service Behavior **[DigitalOOSBehavior]**	Determines the method for setting digital trunks to Out-Of-Service state per device. For a description, refer to the parameter DigitalOOSBehaviorFor Trunk_ID. **Note:** To configure the method for setting Out-Of-Service state per trunk, use the parameter DigitalOOSBehaviorForTrunk_ID.
Web: Default Cause Mapping From ISDN to SIP **[DefaultCauseMapISDN2IP]**	Defines a single default ISDN release cause that is used (in ISDN-to-IP calls) instead of all received release causes, except when the following Q.931 cause values are received: Normal Call Clearing (16), User Busy (17), No User Responding (18), or No Answer from User (19). The range is any valid Q.931 release cause (0 to 127). The default value is 0 (i.e., not configured - static mapping is used).
Web: Release Cause Mapping Table **EMS: ISDN to SIP Cause Mapping**	
[CauseMapISDN2SIP]	This *ini* file table parameter maps ISDN Q.850 Release Causes to SIP responses. The format of this parameter is as follows: [CauseMapISDN2SIP] FORMAT CauseMapISDN2SIP_Index = CauseMapISDN2SIP_IsdnReleaseCause, CauseMapISDN2SIP_SipResponse; [\CauseMapISDN2SIP] Where, ▪ IsdnReleaseCause = Q.850 Release Cause ▪ SipResponse = SIP Response For example: CauseMapISDN2SIP 0 = 50,480; CauseMapISDN2SIP 0 = 6,406;

Parameter	Description
	When a Release Cause is received (from the PSTN side), the device searches this mapping table for a match. If the Q.850 Release Cause is found, the SIP response assigned to it is sent to the IP side. If no match is found, the default static mapping is used. **Notes:** ▪ This parameter can appear up to 12 times. ▪ For an explanation on *ini* file table parameters, refer to "Configuring *ini* File Table Parameters" on page 222.
Web: Release Cause Mapping Table **EMS: SIP to ISDN Cause Mapping**	
[CauseMapSIP2ISDN]	This *ini* file table parameter maps SIP responses to Q.850 Release Causes. The format of this parameter is as follows: [CauseMapSIP2ISDN] FORMAT CauseMapSIP2ISDN_Index = CauseMapSIP2ISDN_SipResponse, CauseMapSIP2ISDN_IsdnReleaseCause; [\CauseMapSIP2ISDN] Where, ▪ SipResponse = SIP Response ▪ IsdnReleaseCause = Q.850 Release Cause For example: CauseMapSIP2ISDN 0 = 480,50; CauseMapSIP2ISDN 0 = 404,3; When a SIP response is received (from the IP side), the device searches this mapping table for a match. If the SIP response is found, the Q.850 Release Cause assigned to it is sent to the PSTN. If no match is found, the default static mapping is used. **Notes:** ▪ This parameter can appear up to 12 times. ▪ For an explanation on *ini* file table parameters, refer to "Configuring *ini* File Table Parameters" on page 222.
Web/EMS: Enable Calling Party Category **[EnableCallingPartyCategory]**	Determines whether Calling Party Category (CPC) is mapped between SIP and PRI. ▪ **[0]** Disable = Don't relay the CPC between SIP and PRI (default). ▪ **[1]** Enable = The CPC is relayed between SIP and PRI. If enabled, the CPC received in the Originating Line Information (OLI) IE of an incoming ISDN Setup message is relayed to the From/P-Asserted-Identity headers using the 'cpc' parameter in the outgoing INVITE message, and vice versa. For example (calling party is a payphone): From:<sip:2000;cpc=payphone@10.8.23.70>;tag=1c1806157451 **Note:** This feature is applicable only to the NI-2 PRI variant.

Parameter	Description
[UserToUserHeaderFormat]	Determines the format of the User-to-User SIP header in the INVITE message for interworking the ISDN User to User (UU) IE data to SIP. ▪ **[0]** = Format: X-UserToUser (default). ▪ **[1]** = Format: User-to-User with Protocol Discriminator (pd) attribute. User-to-User=30303734353137343136353533b313233343b3834;pd=4. (This format is according to "draft-johnston-sipping-cc-uui-04".) ▪ **[2]** = Format: User-to-User with encoding=hex at the end and pd embedded as the first byte. User-to-User=043030373435313734313635353b313233343b3834; encoding=hex. Where "04" at the beginning of this message is the pd. (This format is according to "draft-johnston-sipping-cc-uui-03".)
Web/EMS: Remove CLI when Restricted **[RemoveCLIWhenRestricted]**	Determines (for IP-to-Tel calls) whether the Calling Number and Calling Name IEs are removed from the ISDN Setup message if the presentation is set to Restricted. ▪ **[0]** No = IE's are not removed (default). ▪ **[1]** Yes = IE's are removed.
Web/EMS: Remove Calling Name **[RemoveCallingName]**	Enables the device to remove the Calling Name from SIP-to-ISDN calls for all trunks. ▪ **[0]** Disable = Does not remove Calling Name (default). ▪ **[1]** Enable = Removes Calling Name.
Web: Remove Calling Name EMS: Remove Calling Name For Trunk Mode **[RemoveCallingNameForTrunk_ID]**	Enables the device to remove the Calling Name per trunk (where ID denotes the trunk number) for SIP-to-ISDN calls. ▪ **[-1]** Use Global Parameter = Settings of the global parameter RemoveCallingName are used (default). ▪ **[0]** Disable = Does not remove Calling Name. ▪ **[1]** Enable = Remove Calling Name.
Web/EMS: Progress Indicator to ISDN **[ProgressIndicator2ISDN_ID]**	Progress Indicator (PI) to ISDN. The *ID* in the *ini* file parameter depicts the trunk number, where 0 is the first trunk. ▪ **[-1]** Not Configured = The PI in ISDN messages is set according to the parameter PlayRBTone2Tel (default). ▪ **[0]** No PI = PI is not sent to ISDN. ▪ **[1]** PI = 1; **[8]** PI = 8: The PI value is sent to PSTN in Q.931/Proceeding and Alerting messages. Typically, the PSTN/PBX cuts through the audio channel without playing local Ringback tone, enabling the originating party to hear remote Call Progress Tones or network announcements.
Web: Set PI in Rx Disconnect Message EMS: Set PI For Disconnect Msg **[PIForDisconnectMsg_ID]**	Defines the device's behavior when a Disconnect message is received from the ISDN before a Connect message is received. The *ID* in the *ini* file parameter depicts the trunk number, where 0 is the first trunk. ▪ **[-1]** Not Configured = Sends a 183 SIP response according to the received progress indicator (PI) in the ISDN Disconnect message. If PI = 1 or 8, the device

Parameter	Description
	sends a 183 response, enabling the PSTN to play a voice announcement to the IP side. If there isn't a PI in the Disconnect message, the call is released (default). ▪ **[0]** No PI = Doesn't send a 183 response to IP. The call is released. ▪ **[1]** PI = 1; **[8]** PI = 8: Sends a 183 response to IP.
EMS: Connect On Progress Ind **[ConnectOnProgressInd]**	Enables the play of announcements from IP to PSTN without the need to answer the Tel-to-IP call. It can be used with PSTN networks that don't support the opening of a TDM channel before an ISDN Connect message is received. ▪ **[0]** = Connect message isn't sent after SIP 183 Session Progress message is received (default). ▪ **[1]** = Connect message is sent after SIP 183 Session Progress message is received.
Web: Local ISDN Ringback Tone Source EMS: Local ISDN RB Source **[LocalISDNRBSource_ID]**	Determines whether the Ringback tone is played to the ISDN by the PBX/PSTN or by the device. ▪ **[0]** PBX = PBX/PSTN (default). ▪ **[1]** Gateway = device plays the Ringback tone. This parameter is applicable to ISDN protocols. It is used simultaneously with the parameter PlayRBTone2Trunk. The *ID* in the *ini* file parameter depicts the trunk number, where 0 is the first trunk.
Web: PSTN Alert Timeout EMS: Trunk PSTN Alert Timeout **[PSTNAlertTimeout]**	For digital interfaces: Alert Timeout (in seconds) (ISDN T301 timer) for calls to PSTN. This timer is used between the time a Setup message is sent to the Tel side (IP-to-Tel call establishment) and a Connect message is received. If an Alerting message is received, the timer is restarted. For analog interfaces: Alert Timeout (in seconds) for calls to the Tel side. This timer is used between the time ring is generated (FXS) or line is seized (FXO) until the call is connected. The range is 1 to 600. The default is 180 seconds. **Note:** If per trunk configuration (using TrunkPSTNAlertTimeout) is set to other than default, the PSTNAlertTimeout parameter value is overridden.
Web/EMS: PSTN Alert Timeout **[TrunkPSTNAlertTimeout_ID]**	Alert Timeout (ISDN T301 timer) in seconds for outgoing calls to PSTN. This timer is used between the time that an ISDN Setup message is sent to the Tel side (IP-to-Tel call establishment) and a Connect message is received. If Alerting is received, the timer is restarted. In the *ini* file parameter, *ID* depicts the trunk number, where 0 is the first trunk. The range is 1 to 600. The default is 180.
Web: B-Channel Negotiation EMS: B-Channel Negotiation For Trunk Mode **[BChannelNegotiationForTrunk_ID]**	Determines the ISDN B-channel negotiation mode. ▪ **[-1]** Not Configured = use per device configuration of the BChannelNegotiation parameter (default). ▪ **[0]** Preferred = Preferred. ▪ **[1]** Exclusive = Exclusive. ▪ **[2]** Any = Any.

Parameter	Description
	Notes: ▪ This parameter is applicable to ISDN protocols. ▪ The option 'Any' is only applicable if TerminationSide is set to 0 (i.e., User side). ▪ The *ID* in the *ini* file parameter name represents the trunk number, where 0 is the first trunk.
EMS: Support Redirect InFacility **[SupportRedirectInFacility]**	Determines whether the Redirect Number is retrieved from the Facility IE. ▪ **[0]** = Not supported (default). ▪ **[1]** = Supports partial retrieval of Redirect Number (number only) from the Facility IE in ISDN Setup messages. This is applicable to Redirect Number according to ECMA-173 Call Diversion Supplementary Services. **Note:** To enable this feature, the parameter ISDNDuplicateQ931BuffMode must be set to 1.
[CallReroutingMode]	Determines whether ISDN call rerouting (call forward) is performed by the PSTN instead of the SIP side. This call forwarding is based on Call Deflection for Euro ISDN (ETS-300-207-1) and QSIG (ETSI TS 102 393). ▪ **[0]** Disable (default) ▪ **[1]** Enable = Enables ISDN call rerouting. When the device sends the INVITE message to the remote SIP entity and receives a SIP 302 response with a Contact header containing a URI host name that is the same as the device's IP address, the device sends a Facility message with a Call Rerouting invoke method to the ISDN and waits for the PSTN side to disconnect the call. **Note:** When this parameter is enabled, ensure that the following is configured in the 'Inbound IP Routing Table' (PSTNPrefix *ini* file parameter): ▪ In the 'Destination Phone Prefix' field, enter the original PSTN destination number. ▪ In the 'Source IP Address' field, enter the device's IP address. ▪ Configure the Trunk Group ID. The ISDN call rerouting occurs only if the destination Trunk Group ID is the same as the Trunk Group from where the call was received.
EMS: Enable CIC **[EnableCIC]**	Determines whether the Carrier Identification Code (CIC) is relayed to ISDN. ▪ **[0]** = Do not relay the Carrier Identification Code (CIC) to ISDN (default). ▪ **[1]** = CIC is relayed to the ISDN in Transit Network Selection (TNS) IE. If enabled, the CIC code (received in an INVITE Request-URI) is included in a TNS IE in the ISDN Setup message. For example: INVITE sip:555666;cic=2345@100.2.3.4 sip/2.0.

Parameter	Description
	Notes: ▪ This feature is supported only for SIP-to-ISDN calls. ▪ The parameter AddCicAsPrefix can be used to add the CIC as a prefix to the destination phone number for routing IP-to-Tel calls.
EMS: Enable AOC **[EnableAOC]**	Determines whether ISDN Advice of Charge (AOC) messages are interworked to SIP. ▪ **[0]** = Not used (default). ▪ **[1]** = AOC messages are interworked to SIP. The device supports the receipt of ISDN (Euro ISDN) AOC messages. AOC messages can be received during a call (Facility messages) or at the end of a call (Disconnect or Release messages). The device converts the AOC messages into SIP INFO (during a call) and BYE (end of a call) messages, using a proprietary AOC SIP header. The device supports both Currency and Pulse AOC messages.
EMS: DSP Detectors Enable **[EnableDSPIPMDetectors]**	Enables or disables the device's DSP detectors. ▪ **[0]** = Disable (default). ▪ **[1]** = Enable. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ The device's Software Upgrade Key must contain the 'IPMDetector' DSP option. ▪ When enabled (1), the number of available channels is reduced.
Web: Add IE in SETUP EMS: IE To Be Added In Q.931 Setup **[AddIEinSetup]**	Adds an optional Information Element (IE) data (in hex format) to ISDN Setup messages. For example, to add IE '0x20,0x02,0x00,0xe1', enter the value "200200e1". **Notes:** ▪ This IE is sent from the Trunk Group IDs that are defined by the parameter SendIEonTG. ▪ You can configure different IE data for Trunk Groups by defining this parameter for different IP Profile IDs (using the parameter IPProfile) and then assigning the required IP Profile ID in the 'Inbound IP Routing Table' (PSTNPrefix).
Web: Trunk Groups to Send IE EMS: List Of Trunk Groups To Send IE **[SendIEonTG]**	Defines Trunk Group IDs (up to 50 characters) from where the optional ISDN IE (defined by the parameter AddIEinSetup) is sent. For example: '1,2,4,10,12,6'. **Notes:** ▪ You can configure different IE data for Trunk Groups by defining this parameter for different IP Profile IDs (using the parameter IPProfile), and then assigning the required IP Profile ID in the 'Inbound IP Routing Table' (PSTNPrefix).

Parameter	Description
	▪ When IP Profiles are used for configuring different IE data for Trunk Groups, this parameter is ignored.
Web: Enable User-to-User IE for Tel to IP EMS: Enable UUI Tel 2 Ip **[EnableUUITel2IP]**	Enables ISDN PRI-to-SIP interworking. ▪ **[0]** Disable = Disabled (default). ▪ **[1]** Enable = Enable transfer of User-to-User (UU) IE from PRI to SIP. The device supports the following ISDN PRI-to-SIP interworking: Setup to SIP INVITE, Connect to SIP 200 OK, User Information to SIP INFO, Alerting to SIP 18x response, and Disconnect to SIP BYE response messages. **Note:** The interworking of ISDN User-to-User IE to SIP INFO is applicable only to the Euro ISDN, QSIG, and 4ESS PRI variants.
Web: Enable User-to-User IE for IP to Tel EMS: Enable UUI Ip 2 Tel **[EnableUUIIP2Tel]**	Enables SIP-to-PRI ISDN interworking. ▪ **[0]** Disable = Disabled (default). ▪ **[1]** Enable = Enable transfer of User-to-User (UU) IE from SIP INVITE message to PRI Setup message. The device supports the following SIP-to-PRI ISDN interworking: SIP INVITE to Setup, SIP 200 OK to Connect, SIP INFO to User Information, SIP 18x to Alerting, and SIP BYE to Disconnect. **Notes:** ▪ The interworking of ISDN User-to-User IE to SIP INFO is applicable only to the Euro ISDN, QSIG, and 4ESS PRI variants. ▪ To interwork the UUIE header from SIP-to-ISDN messages with the 4ESS ISDN variant, the parameter ISDNGeneralCCBehavior must be set to 16384.
[Enable911LocationIdIP2Tel]	Enables interworking of Emergency Location Identification from SIP to PRI. ▪ **[0]** = Disabled (default) ▪ **[1]** = Enabled When enabled, the From header received in the SIP INVITE is translated into the following ISDN IE's: ▪ Emergency Call Control. ▪ Generic Information - to carry the Location Identification Number information. ▪ Generic Information - to carry the Calling Geodetic Location information. **Note:** This capability is applicable only to the NI-2 ISDN variant.
[EarlyAnswerTimeout]	Defines the time (in seconds) that the device waits for an ISDN Connect message from the called party (Tel side) after sending a Setup message. If the timer expires, the call is answered by sending a SIP 200 OK message (IP side). The valid range is 0 to 600. The default value is 0 (i.e., disabled).

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>Web/EMS: Trunk Transfer Mode
[TrunkTransferMode]</td><td>Determines the trunk transfer method (for all trunks) when a SIP REFER message is received. The transfer method depends on the Trunk's PSTN protocol (configured by the parameter ProtocolType) and is applicable only when one of these protocols are used:
<table>
<tr><th>PSTN Protocol</th><th>Transfer Method</th></tr>
<tr><td>E1 Euro ISDN [1]</td><td>ECT [2] or InBand [5]</td></tr>
<tr><td>E1 QSIG [21],
T1 QSIG [23]</td><td>Single Step Transfer [4], Path Replacement Transfer [2], or InBand [5]</td></tr>
<tr><td>T1 NI2 ISDN [10],
T1 4ESS ISDN [11],
T1 5ESS 9 ISDN [12]</td><td>TBCT [2] or InBand [5]</td></tr>
<tr><td>T1 DMS100 ISDN [14]</td><td>RTL [2] or InBand [5]</td></tr>
<tr><td>T1 RAW CAS [3], T1 CAS [2], E1 CAS [8], E1 RAW CAS [9]</td><td>[1] CAS NFA DMS-100 or [3] CAS Normal transfer</td></tr>
</table>
The valid values of this parameter are described below:

[0] = Not supported (default).
[1] = Supports CAS NFA DMS-100 transfer. When a SIP REFER message is received, the device performs a Blind Transfer by executing a CAS Wink, waits for an acknowledged Wink from the remote side, dials the Refer-to number to the switch, and then releases the call.
Note: A specific NFA CAS table is required.
[2] = Supports ISDN (PRI and BRI) transfer - Release Link Trunk (RLT) (DMS-100), Two B Channel Transfer (TBCT) (NI2), Explicit Call Transfer (ECT) (EURO ISDN), and Path Replacement (QSIG). When a SIP REFER message is received, the device performs a transfer by sending Facility messages to the PBX with the necessary information on the call's legs to be connected. The different ISDN variants use slightly different methods (using Facility messages) to perform the transfer. Notes:

✓ For RLT ISDN transfer, the parameter SendISDNTransferOnConnect must be set to 1.
✓ The parameter SendISDNTransferOnConnect can be used to define if the TBCT/ECT transfer is performed after receipt of Alerting or Connect messages. For RLT, the transfer is always done after receipt of Connect (SendISDNTransferOnConnect is set to 1).
✓ This transfer can be performed between B-channels from different trunks or Trunk Groups, by using the parameter EnableTransferAcrossTrunkGroups.

[3] = Supports CAS Normal transfer. When a SIP REFER message is received, the device performs a Blind Transfer by executing a CAS Wink, dialing the Refer-to number to the switch, and then releasing the call.
[4] = Supports QSIG Single Step transfer (PRI and BRI): IP-to-Tel: When a SIP REFER message is received, the
</td></tr>
</table>

Parameter	Description
	device performs a transfer by sending a Facility message to the PBX, initiating Single Step transfer. Once a success return result is received, the transfer is completed. Tel-to-IP: When a Facility message initiating Single Step transfer is received from the PBX, a SIP REFER message is sent to the IP side. ▪ **[5]** = IP-to-Tel Blind Transfer mode supported for ISDN (PRI and BRI) protocols and implemented according to AT&T Toll Free Transfer Connect Service (TR 50075) "Courtesy Transfer-Human-No Data". When the device receives a SIP REFER message, it performs a blind transfer by first dialing the DTMF digits (transfer prefix) defined by the parameter XferPrefixIP2Tel (configured to "*8" for AT&T service), and then (after 500 msec) the device dials the DTMF of the number (referred) from the Refer-To header sip:URI userpart. If the hostpart of the Refer-To sip:URI contains the device's IP address, and if the Trunk Group selected according to the IP to Tel Routing table is the same Trunk Group as the original call, then the device performs the in-band DTMF transfer; otherwise, the device sends the INVITE according to regular transfer rules. After completing the in-band transfer, the device waits for the ISDN Disconnect message. If the Disconnect message is received during the first 5 seconds, the device sends a SIP NOTIFY with 200 OK message; otherwise, the device sends a NOTIFY with 4xx message. **Note:** For configuring trunk transfer mode per trunk, use the parameter TrunkTransferMode_X.
[TrunkTransferMode_X]	Determines the trunk transfer mode per trunk (where *x* is the Trunk ID). For configuring trunk transfer mode for all trunks and for a description of the parameter options, refer to the parameter TrunkTransferMode.
[EnableTransferAcrossTrunkGroups]	Determines whether the device allows ISDN ECT, RLT or TBCT IP-to-Tel call transfers between B-channels of different Trunk Groups. ▪ **[0]** = Disable - ISDN call transfer is only between B-channels of the same Trunk Group (default). ▪ **[1]** = Enable - the device performs ISDN transfer between any two PSTN calls (between any Trunk Group) handled by the device. **Note:** The ISDN transfer also requires that you configure the parameter TrunkTransferMode_x to 2.

Parameter	Description
Web: ISDN Transfer Capabilities EMS: Transfer Capability To ISDN **[ISDNTransferCapability_ID]**	Defines the IP-to-ISDN Transfer Capability of the Bearer Capability IE in ISDN Setup messages. The *ID* in the *ini* file parameter depicts the trunk number, where 0 is the first trunk. ▪ **[-1]** Not Configured ▪ **[0]** Audio 3.1 = Audio (default). ▪ **[1]** Speech = Speech. ▪ **[2]** Data = Data. ▪ Audio 7 = Currently not supported. **Note:** If this parameter isn't configured or equals to '-1', Audio 3.1 capability is used.
Web: ISDN Transfer On Connect EMS: Send ISDN Transfer On Connect **[SendISDNTransferOnConnect]**	This parameter is used for the ECT/TBCT/RLT/Path Replacement ISDN transfer methods. Usually, the device requests the PBX to connect an incoming and outgoing call. This parameter determines if the outgoing call (from the device to the PBX) must be connected before the transfer is initiated. ▪ **[0]** Alert = Enables ISDN Transfer if the outgoing call is in Alerting or Connect state (default). ▪ **[1]** Connect = Enables ISDN Transfer only if the outgoing call is in Connect state. **Note:** For RLT ISDN transfer (TrunkTransferMode = 2 and ProtocolType = 14 DMS-100), this parameter must be set to 1.
[ISDNTransferCompleteTimeout]	The timeout (in seconds) for determining ISDN call transfer (ECT, RLT, or TBCT) failure. If the device does not receive any response to an ISDN transfer attempt within this user-defined time, the device identifies this as an ISDN transfer failure and subsequently performs a hairpin TDM connection or sends a SIP NOTIFY message with a SIP 603 response (depending whether hairpin is enabled or disabled, using the parameter DisableFallbackTransferToTDM). The valid range is 1 to 10. The default is 4.
Web/EMS: Enable Network ISDN Transfer **[EnableNetworkISDNTransfer]**	Determines whether the device allows network-side ISDN transfer requests for IP-to-ISDN calls. These ISDN path replacements include NI2 TBCT (Two B-channel Transfer) and ETSI ECT (Explicit Call Transfer). ▪ **[0]** Disable = Rejects ISDN transfer requests. ▪ **[1]** Enable (default) = The device sends a SIP REFER message to the remote call party if such a path replacement is received from the ISDN side (e.g., from a PBX).
[DisableFallbackTransferToTDM]	Enables or disables "hairpin" TDM transfer upon ISDN (ECT, RLT, or TBCT) call transfer failure. When this feature is enabled and an ISDN call transfer failure occurs, the device sends a SIP NOTIFY message with a SIP 603 Decline response. ▪ **[0]** = device performs a hairpin TDM transfer upon ISDN call transfer (default). ▪ **[1]** = Hairpin TDM transfer is disabled.

Parameter	Description
Web: Enable QSIG Transfer Update **[EnableQSIGTransferUpdate]**	Determines whether the device interworks QSIG Facility messages with callTranferComplete invoke application protocol data unit (APDU) to SIP UPDATE messages with P-Asserted-Identity and optional Privacy headers. This feature is supported for IP-to-Tel and Tel-to-IP calls. ▪ **[0]** Disable (default) = Ignores QSIG Facility message with callTranferComplete invoke ▪ **[1]** Enable For example, assume A and C are PBX call parties, and B is the SIP IP phone: 1 A calls B; B answers the call. 2 A places B on hold, and calls C; C answers the call. 3 A performs a call transfer (the transfer is done internally by the PBX); B and C are connected to one another. In the above example, the PBX updates B that it is now talking with C. The PBX updates this by sending a QSIG Facility message with callTranferComplete invoke APDU. The device interworks this message to a SIP UPDATE message containing a P-Asserted-Identity header with the number and name derived from QSIG callTranferComplete redirectionNumber and redirectionName. **Note:** For IP-to-Tel calls, the redirectionNumber and redirectionName in the callTRansferComplete invoke is derived from the P-Asserted-Identity and Privacy headers.
[CASSendHookFlash]	Enables sending Wink signal toward CAS trunks. ▪ **[0]** = Disable (default). ▪ **[1]** = Enable. If the device receives a mid-call SIP INFO message with flashhook event body (as shown below) and this parameter is set to 1, the device generates a wink signal toward the CAS trunk. The CAS wink signal is done by changing the A bit from 1 to 0, and then back to 1 for 450 msec. INFO sip:4505656002@192.168.13.40:5060 SIP/2.0 Via: SIP/2.0/UDP 192.168.13.2:5060 From: <sip:06@192.168.13.2:5060> To: <sip:4505656002@192.168.13.40:5060>;tag=132878796-1040067870294 Call-ID: 0010-0016-D69A7DA8-1@192.168.13.2 CSeq:2 INFO Content-Type: application/broadsoft Content-Length: 17 event flashhook **Note:** This parameter is applicable only to T1 CAS protocols.

6.13 Answer and Disconnect Supervision Parameters

The answer and disconnect supervision parameters are described in the table below.

Table 6-54: Answer and Disconnect Parameters

Parameter	Description
Web: Answer Supervision EMS: Enable Voice Detection **[EnableVoiceDetection]**	Enables the sending of SIP 200 OK upon detection of speech, fax, or modem. ▪ **[1]** Yes = The device sends SIP 200 OK (to INVITE) messages when speech/fax/modem is detected from the Tel side. ▪ **[0]** No = The device sends SIP 200 OK only after it completes dialing to the Tel side (default). Typically, this feature is used only when early media (EnableEarlyMedia) is used to establish the voice path before the call is answered. **Notes:** ▪ FXO interfaces: This feature is applicable only to one-stage dialing (FXO). ▪ Digital interfaces: To activate this feature, set the parameter EnableDSPIPMDetectors to 1. ▪ Digital interfaces: This feature is applicable only when the protocol type is CAS.
Web/EMS: Max Call Duration (min) **[MaxCallDuration]**	Defines the maximum call duration (in minutes). If this time expires, both sides of the call are released (IP and Tel). The valid range is 0 to 35,791. The default is 0 (i.e., no limitation).
Web/EMS: Disconnect on Dial Tone **[DisconnectOnDialTone]**	Determines whether the device disconnects a call when a dial tone is detected from the PBX. ▪ **[0]** Disable = Call is not released (default). ▪ **[1]** Enable = Call is released if dial tone is detected on the device's FXO port. **Notes:** ▪ This parameter is applicable only to FXO interfaces. ▪ This option is in addition to the mechanism that disconnects a call when either busy or reorder tones are detected.
Web: Send Digit Pattern on Connect EMS: Connect Code **[TelConnectCode]**	Defines a digit pattern to send to the Tel side after a SIP 200 OK is received from the IP side. The digit pattern is a user-defined DTMF sequence that is used to indicate an answer signal (e.g., for billing). The valid range is 1 to 8 characters. **Note:** This parameter is applicable to FXO/CAS.

Parameter	Description
Web: Disconnect on Broken Connection EMS: Disconnect Calls on Broken Connection **[DisconnectOnBrokenConnection]**	Determines whether the device releases the call if RTP packets are not received within a user-defined timeout. ▪ **[0]** No ▪ **[1]** Yes (default) **Notes:** ▪ The timeout is configured by the parameter BrokenConnectionEventTimeout. ▪ This feature is applicable only if the RTP session is used without Silence Compression. If Silence Compression is enabled, the device doesn't detect a broken RTP connection. ▪ During a call, if the source IP address (from where the RTP packets are received) is changed without notifying the device, the device filters these RTP packets. To overcome this, set the parameter DisconnectOnBrokenConnection to 0; the device doesn't detect RTP packets arriving from the original source IP address and switches (after 300 msec) to the RTP packets arriving from the new source IP address.
Web: Broken Connection Timeout EMS: Broken Connection Event Timeout **[BrokenConnectionEventTimeout]**	The time period (in 100 msec units) after which a call is disconnected if an RTP packet is not received. The valid range is 1 to 1,000. The default value is 100 (i.e., 10 seconds). **Notes:** ▪ This parameter is applicable only if the parameter DisconnectOnBrokenConnection is set to 1. ▪ Currently, this feature functions only if Silence Suppression is disabled.
Web: Disconnect Call on Silence Detection EMS: Disconnect On Detection Of Silence **[EnableSilenceDisconnect]**	Determines whether calls are disconnected after detection of silence. ▪ **[1]** Yes = The device disconnects calls in which silence occurs (in both call directions) for more than a user-defined time. ▪ **[0]** No = Call is not disconnected when silence is detected (default). The silence duration can be configured by the FarEndDisconnectSilencePeriod parameter (default 120). **Note:** To activate this feature, set the parameters EnableSilenceCompression and FarEndDisconnectSilenceMethod to 1.
Web: Silence Detection Period **[sec]** EMS: Silence Detection Time Out **[FarEndDisconnectSilencePeriod]**	Duration of the silence period (in seconds) after which the call is disconnected. The range is 10 to 28,800 (i.e., 8 hours). The default is 120 seconds. **Note:** For this parameter to take effect, a device reset is required.

Parameter	Description
Web: Silence Detection Method **[FarEndDisconnectSilenceMethod]**	Silence detection method. ▪ **[0]** None = Silence detection option is disabled. ▪ **[1]** Packets Count = According to packet count. ▪ **[2]** Voice/Energy Detectors = N/A. ▪ **[3]** All = N/A. **Note:** For this parameter to take effect, a device reset is required.
[FarEndDisconnectSilenceThreshold]	Threshold of the packet count (in percentages) below which is considered silence by the device. The valid range is 1 to 100%. The default is 8%. **Notes:** ▪ This parameter is applicable only if silence is detected according to packet count (FarEndDisconnectSilenceMethod is set to 1). ▪ For this parameter to take effect, a device reset is required.
[BrokenConnectionDuringSilence]	Enables the generation of the BrokenConnection event during a silence period if the channel's NoOp feature is enabled (using the parameter NoOpEnable) and if the channel stops receiving NoOp RTP packets. ▪ **[0]** Disable (default). ▪ **[1]** Enable.
Web: Trunk Alarm Call Disconnect Timeout **[TrunkAlarmCallDisconnectTimeout]**	Time in seconds to wait (in seconds) after an E1/T1 trunk "red" alarm (LOS/LOF) is raised before the device disconnects the SIP call. Once this user-defined time elapses, the device sends a SIP BYE message to terminate the call. If the alarm is cleared before this timeout elapses, the call is not terminated and continues as normal. The range is 1 to 80. The default is 0 (20 for E1 and 40 for T1).
Web: Disconnect Call on Busy Tone Detection (ISDN) EMS: Isdn Disconnect On Busy Tone **[ISDNDisconnectOnBusyTone]**	Determines whether a call is disconnected upon detection of a busy tone (for ISDN). ▪ **[0]** Disable = Do not disconnect call upon detection of busy tone. ▪ **[1]** Enable = Disconnect call upon detection of busy tone (default). **Notes:** ▪ This parameter is applicable only to ISDN protocols. ▪ IP-to-ISDN calls are disconnected on detection of SIT tones only in call alert state. If the call is in connected state, the SIT does not disconnect the calls. Detection of Busy or Reorder tones disconnect the IP-to-ISDN calls also in call connected state. ▪ For IP-to-CAS calls, detection of Busy, Reorder or SIT tones disconnect the calls in any call state.

Parameter	Description
Web: Disconnect Call on Busy Tone Detection (CAS) EMS: Disconnect On Detection End Tones **[DisconnectOnBusyTone]**	Determines whether a call is disconnected upon detection of a busy tone (for CAS). ▪ **[0]** Disable = Do not disconnect call on detection of busy tone. ▪ **[1]** Enable = Call is released if busy or reorder (fast busy) tone is detected on the device's FXO port (default). **Notes:** ▪ Digital interfaces: This parameter is applicable only to CAS protocols. ▪ Analog interfaces: This parameter is applicable only to FXO interfaces.
Polarity (Current) Reversal for Call Release (Analog Interfaces) Parameters	
Web: Enable Polarity Reversal EMS: Enable Reversal Polarity **[EnableReversalPolarity]**	Enables the polarity reversal feature for call release. ▪ **[0]** Disable = Disable the polarity reversal service (default). ▪ **[1]** Enable = Enable the polarity reversal service. If the polarity reversal service is enabled, the FXS interface changes the line polarity on call answer and then changes it back on call release. The FXO interface sends a 200 OK response when polarity reversal signal is detected (applicable only to one-stage dialing) and releases a call when a second polarity reversal signal is detected.
Web/EMS: Enable Current Disconnect **[EnableCurrentDisconnect]**	Enables call release upon detection of a Current Disconnect signal. ▪ **[0]** Disable = Disable the current disconnect service (default). ▪ **[1]** Enable = Enable the current disconnect service. If the current disconnect service is enabled: ▪ The FXO releases a call when a current disconnect signal is detected on its port. ▪ The FXS interface generates a 'Current Disconnect Pulse' after a call is released from IP. The current disconnect duration is determined by the parameter CurrentDisconnectDuration. The current disconnect threshold (FXO only) is determined by the parameter CurrentDisconnectDefaultThreshold. The frequency at which the analog line voltage is sampled is determined by the parameter TimeToSampleAnalogLineVoltage.

Parameter	Description
EMS: Polarity Reversal Type **[PolarityReversalType]**	Defines the voltage change slope during polarity reversal or wink. ▪ **[0]** = Soft reverse polarity (default). ▪ **[1]** = Hard reverse polarity. **Notes:** ▪ This parameter is applicable only to FXS interfaces. ▪ Some Caller ID signals use reversal polarity and/or Wink signals. In these cases, it is recommended to set the parameter PolarityReversalType to 1 (Hard). ▪ For this parameter to take effect, a device reset is required.
EMS: Current Disconnect Duration **[CurrentDisconnectDuration]**	The duration (in msec) of the current disconnect pulse. The range is 200 to 1500. The default is 900. **Notes:** ▪ This parameter is applicable for FXS and FXO interfaces. ▪ The FXO interface detection window is 100 msec below the parameter's value and 350 msec above the parameter's value. For example, if this parameter is set to 400 msec, then the detection window is 300 to 750 msec. ▪ For this parameter to take effect, a device reset is required.
[CurrentDisconnectDefaultThreshold]	Determines the line voltage threshold at which a current disconnect detection is considered. The valid range is 0 to 20 Volts. The default value is 4 Volts. **Notes:** ▪ This parameter is applicable only to FXO interfaces. ▪ For this parameter to take effect, a device reset is required.
[TimeToSampleAnalogLineVoltage]	Determines the frequency at which the analog line voltage is sampled (after offhook), for detection of the current disconnect threshold. The valid range is 100 to 2500 msec. The default value is 1000 msec. **Notes:** ▪ This parameter is applicable only to FXO interfaces. ▪ For this parameter to take effect, a device reset is required.

6.14 Tone Parameters

This subsection describes the device's tone parameters.

6.14.1 Telephony Tone Parameters

The telephony tone parameters are described in the table below.

Table 6-55: Tone Parameters

Parameter	Description
Tone Index Table	
[ToneIndex]	This *ini* file table parameter configures the Tone Index table, which allows you to define Distinctive Ringing and Call Waiting tones per FXS endpoint (or for a range of FXS endpoints), and is based on calling number (source number prefix) for IP-to-Tel calls. This allows different tones to be played for an FXS endpoint depending on the source number of the received call. The format of this parameter is as follows: [ToneIndex] FORMAT ToneIndex_Index = ToneIndex_FXSPort_First, ToneIndex_FXSPort_Last, ToneIndex_SourcePrefix, ToneIndex_PriorityIndex; [\ToneIndex] Where, ▪ FXSPort_First = starting range of FXS ports. ▪ FXSPort_Last = end range of FXS ports. ▪ SourcePrefix = prefix of the calling number. ▪ PriorityIndex = index for Distinctive Ringing and Call Waiting tones (default is 0): ✓ Ringing tone index = index in the CPT file for playing the ring tone. ✓ Call Waiting tone index = priority index + FirstCallWaitingToneID(*). For example, if you want to select the Call Waiting tone defined in the CPT file at Index #9, then you can enter 1 as the priority index and the value 8 for FirstCallWaitingToneID. The summation of these values equals 9, i.e., index #9. For example, the configuration below plays the tone Index #3 to FXS ports 1 and 2 if the source number prefix of the received call is 20. ToneIndex 1 = 1, 2, 20*, 3; **Notes:** ▪ You can define up to 50 indices. ▪ This parameter is applicable only to FXS interfaces. ▪ Typically, the Ringing and/or Call Waiting tone played is indicated in the SIP Alert-Info header field of the received INVITE message. If this header is not present, then the tone played is according to the settings in this table. ▪ For depicting a range of FXS ports, use the syntax x-y (e.g., "1-

Parameter	Description
	4" for ports 1 through 4). ▪ You can configure multiple entries with different source prefixes and tones for the same FXS port.
Web/EMS: Dial Tone Duration **[sec]** **[TimeForDialTone]**	Duration (in seconds) that the dial tone is played (for digital interfaces, to an ISDN terminal). For digital interfaces: This parameter is applicable for overlap dialing when ISDNInCallsBehavior is set to 65536. The dial tone is played if the ISDN Setup message doesn't include the called number. The valid range is 0 to 60. The default is 5. For analog interfaces: FXS interfaces play the dial tone after the phone is picked up (off-hook). FXO interfaces play the dial tone after the port is seized in response to ringing (from PBX/PSTN). The valid range is 0 to 60. The default time is 16. Notes for analog interfaces: ▪ During play of dial tone, the device waits for DTMF digits. ▪ This parameter is not applicable when Automatic Dialing is enabled.
Web/EMS: Stutter Tone Duration **[StutterToneDuration]**	Duration (in msec) of the Confirmation tone. A Stutter tone is played (instead of a regular dial tone) when a Message Waiting Indication (MWI) is received. The Stutter tone is composed of a Confirmation tone (Tone Type #8), which is played for the defined duration (StutterToneDuration) followed by a Stutter Dial tone (Tone Type #15). Both these tones are defined in the CPT file. The range is 1,000 to 60,000. The default is 2,000 (i.e., 2 seconds). **Notes:** ▪ This parameter is applicable only to FXS interfaces. ▪ If you want to configure the duration of the Confirmation tone to longer than 16 seconds, you must increase the value of the parameter TimeForDialTone accordingly. ▪ The MWI tone takes precedence over the Call Forwarding Reminder tone. For detailed information on MWI, refer to Message Waiting Indication on page 572.
Web: FXO AutoDial Play BusyTone EMS: Auto Dial Play Busy Tone **[FXOAutoDialPlayBusyTone]**	Determines whether the device plays a Busy/Reorder tone to the PSTN side if a Tel-to-IP call is rejected by a SIP error response (4xx, 5xx or 6xx). If a SIP error response is received, the device seizes the line (off-hook), and then plays a Busy/Reorder tone to the PSTN side (for the duration defined by the parameter TimeForReorderTone). After playing the tone, the line is released (on-hook). ▪ **[0]** = Disable (default) ▪ **[1]** = Enable **Note:** This parameter is applicable only to FXO interfaces.
Web: Hotline Dial Tone Duration EMS: Hot Line Tone Duration **[HotLineToneDuration]**	Duration (in seconds) of the Hotline dial tone. If no digits are received during this duration, the device initiates a call to a user-defined number (refer to Configuring Automatic Dialing on page 172). The valid range is 0 to 60. The default is 16. **Note:** This parameter is applicable to FXS and FXO interfaces.

Parameter	Description
Web: Time Before Reorder Tone **[sec]** EMS: Time For Reorder Tone **[TimeBeforeReorderTone]**	The delay interval (in seconds) from when the device receives a SIP BYE message (i.e., remote party terminates call) until the device starts playing a reorder tone to the FXS phone. The valid range is 0 to 60. The default is 0. **Note:** This parameter is applicable only to FXS interfaces.
Web/EMS: Reorder Tone Duration [sec] **[TimeForReorderTone]**	For Analog: The duration (in seconds) that the device plays a Busy or Reorder tone duration before releasing the line. The valid range is 0 to 254. The default is 0 seconds. Typically, after playing a Reorder/Busy tone for the specified duration, the device starts playing an Offhook Warning tone. For Digital: The duration (in seconds) that the CAS device plays a Busy or Reorder Tone before releasing the line. The valid range is 0 to 254. The default value is 10. **Notes:** ▪ The selection of Busy or Reorder tone is performed according to the release cause received from IP. ▪ For Digital: This parameter is also applicable for ISDN when PlayBusyTone2ISDN is set to 2. ▪ Refer also to the parameter CutThrough.
Web/EMS: Enable Comfort Tone **[EnableComfortTone]**	Determines whether the device plays a Comfort Tone (Tone Type #18) to the FXS/FXO endpoint after a SIP INVITE is sent and before a SIP 18x response is received. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Note:** This parameter is applicable only to FXO/FXS interfaces.
[WarningToneDuration]	Defines the duration (in seconds) for which the Off-Hook Warning Tone is played to the user. The valid range is -1 to 2,147,483,647. The default is 600. **Notes:** ▪ A negative value indicates that the tone is played infinitely. ▪ This parameter is applicable only to analog interfaces.
Web: Play Ringback Tone to Tel EMS: Play Ring Back Tone To Tel **[PlayRBTone2Tel]**	Enables the play of the ringback tone (RBT) to the Tel side and determines the method for playing the RBT. ▪ **[0]** Don't Play = RBT is not played. ▪ **[1]** Play Local = RBT is played to the Tel side of the call when a SIP 180/183 response is received. ▪ **[2]** Play According to Early Media = RBT is played to the Tel side only if a 180/183 response without SDP is received. If 180/183 with SDP message is received, the device cuts through the voice channel and doesn't play RBT (default). ▪ **[3]** Play Local Until Remote Media Arrive = Plays the RBT according to received media. The behaviour is similar to **[2].** If a SIP 180 response is received and the voice channel is already open (due to a previous 183 early media response or due to an SDP in the current 180 response), the device plays a local RBT if there are no prior received RTP packets. The device stops playing the local RBT as soon as it starts receiving RTP packets. At this stage, if the device receives additional 18x responses, it does not resume playing the local RBT.

Parameter	Description
	Note: For ISDN trunks, this option is applicable only if the parameter LocalISDNRBSource is set to 1.
Web: Play Ringback Tone to IP EMS: Play Ring Back Tone To IP **[PlayRBTone2IP]**	Determines whether or not the device plays a ringback tone (RBT) to the IP side for IP-to-Tel calls. ▪ **[0]** Don't Play = Ringback tone isn't played (default). ▪ **[1]** Play = Ringback tone is played after SIP 183 session progress response is sent. For digital modules: If configured to 1 ('Play') and EnableEarlyMedia is set to 1, the device plays a ringback tone according to the following: ▪ For CAS interfaces: the device opens a voice channel, sends a 183+SDP response, and then plays a ringback tone to IP. ▪ For ISDN interfaces: if a Progress or an Alerting message with PI (1 or 8) is received from the ISDN, the device opens a voice channel, sends a 183+SDP or 180+SDP response, but doesn't play a ringback tone to IP. If PI (1 or 8) is received from the ISDN, the device assumes that ringback tone is played by the ISDN switch. Otherwise, the device plays a ringback tone to IP after receiving an Alerting message from the ISDN. It sends a 180+SDP response, signaling to the calling party to open a voice channel to hear the played ringback tone. **Notes:** ▪ To enable the device to send a 183/180+SDP responses, set the parameter EnableEarlyMedia to 1. ▪ If the parameter EnableDigitDelivery is set to 1, the device doesn't play a ringback tone to IP and doesn't send 183 or 180+SDP responses.
Web: Play Local RBT on ISDN Transfer EMS: Play RBT On ISDN Transfer **[PlayRBTOnISDNTransfer]**	Determines whether the device plays a local ringback tone (RBT) for ISDN's Two B Channel Transfer (TBCT), Release Line Trunk (RLT), or Explicit Call Transfer (ECT) call transfers to the originator when the second leg receives an ISDN Alerting or Progress message. ▪ **[0]** Don't Play (default). ▪ **[1]** Play. **Notes:** ▪ For Blind transfer, the local RBT is played to first call PSTN party when the second leg receives the ISDN Alerting or Progress message. ▪ For Consulted transfer, the local RBT is played when the second leg receives ISDN Alerting or Progress message if the Progress message is received after a SIP REFER. ▪ This parameter is applicable only if the parameter SendISDNTransferOnConnect is set to 1.
Web: MFC R2 Category EMS: R2 Category **[R2Category]**	Determines the tone for MFC R2 calling party category (CPC). The parameter provides information on the calling party such as National or International call, Operator or Subscriber and Subscriber priority. The value range is 1 to 15 (defining one of the MFC R2 tones). The default value is 1.

6.14.2 Tone Detection Parameters

The signal tone detection parameters are described in the table below.

Table 6-56: Tone Detection Parameters

Parameter	Description
EMS: DTMF Enable **[DTMFDetectorEnable]**	Enables or disables the detection of DTMF signaling. ▪ **[0]** = Disable ▪ **[1]** = Enable (default)
EMS: MF R1 Enable **[MFR1DetectorEnable]**	Enables or disables the detection of MF-R1 signaling. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable
EMS: R1.5 Detection Standard **[R1DetectionStandard]**	Determines the MF-R1 protocol used for detection. ▪ **[0]** = ITU (default) ▪ **[1]** = R1.5 **Note:** For this parameter to take effect, a device reset is required.
EMS: User Defined Tone Enable **[UserDefinedToneDetectorEnable]**	Enables or disables the detection of User Defined Tones signaling, applicable for Special Information Tone (SIT) detection. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable
EMS: SIT Enable **[SITDetectorEnable]**	Enables or disables SIT detection according to the ITU-T recommendation E.180/Q.35. ▪ **[0]** = Disable (default). ▪ **[1]** = Enable. To disconnect IP-to-ISDN calls when a SIT tone is detected, the following parameters must be configured: ▪ SITDetectorEnable = 1 ▪ UserDefinedToneDetectorEnable = 1 ▪ ISDNDisconnectOnBusyTone = 1 (applicable for Busy, Reorder and SIT tones) Another parameter for handling the SIT tone is SITQ850Cause, which determines the Q.850 cause value specified in the SIP Reason header that is included in a 4xx response when a SIT tone is detected on an IP-to-Tel call. To disconnect IP-to-CAS calls when a SIT tone is detected, the following parameters must be configured (applicable to FXO interfaces): ▪ SITDetectorEnable = 1 ▪ UserDefinedToneDetectorEnable = 1 ▪ DisconnectOnBusyTone = 1 (applicable for Busy, Reorder and SIT tones) **Notes:** ▪ For this parameter to take effect, a device reset is required.

Parameter	Description
	▪ The IP-to-ISDN call is disconnected on detection of a SIT tone only in call alert state. If the call is in connected state, the SIT does not disconnect the call. Detection of Busy or Reorder tones disconnect these calls also in call connected state. ▪ For IP-to-CAS calls, detection of Busy, Reorder, or SIT tones disconnect the call in any call state.
EMS: UDT Detector Frequency Deviation **[UDTDetectorFrequencyDeviation]**	Defines the deviation (in Hz) allowed for the detection of each signal frequency. The valid range is 1 to 50. The default value is 50. **Note:** For this parameter to take effect, a device reset is required.
EMS: CPT Detector Frequency Deviation **[CPTDetectorFrequencyDeviation]**	Defines the deviation (in Hz) allowed for the detection of each CPT signal frequency. The valid range is 1 to 30. The default value is 10. **Note:** For this parameter to take effect, a device reset is required.

6.14.3 Metering Tone Parameters

The metering tone parameters are described in the table below.

Table 6-57: Metering Tone Parameters

Parameter	Description
Web: Generate Metering Tones EMS: Metering Mode **[PayPhoneMeteringMode]**	Determines the method used to configure the metering tones that are generated to the Tel side. ▪ **[0]** Disable = Metering tones aren't generated (default). ▪ **[1]** Internal Table = Metering tones are generated according to the internal table configured by the parameter ChargeCode. **Notes:** ▪ This parameter is applicable only to FXS interfaces. ▪ If you select 'Internal Table', you must configure the 'Charge Codes Table' (refer to "Configuring the Charge Codes Table" on page 139).
Web: Analog Metering Type EMS: Metering Type **[MeteringType]**	Determines the metering method for generating pulses (sinusoidal metering burst frequency) by the FXS port. ▪ **[0]** 12 KHz (default) = 12 kHz sinusoidal bursts ▪ **[1]** 16 KHz = 16 kHz sinusoidal bursts ▪ **[2]** = Polarity Reversal pulses **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ This parameter is applicable only to FXS interfaces.

Parameter	Description
Web: Analog TTX Voltage Level EMS: TTX Voltage Level **[AnalogTTXVoltageLevel]**	Determines the metering signal/pulse voltage level (TTX). ▪ **[0]** = 0 Vrms sinusoidal bursts ▪ **[1]** (default) = 0.5 Vrms sinusoidal bursts ▪ **[2]** = 1 Vrms sinusoidal bursts **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ This parameter is applicable only to FXS interfaces.
Web: Charge Codes Table **EMS: Charge Codes**	
[ChargeCode]	This *ini* file table parameter configures metering tones (and their time intervals) that the device's FXS interface generates to the Tel side. The format of this parameter is as follows: [ChargeCode] FORMAT ChargeCode_Index = ChargeCode_EndTime1, ChargeCode_PulseInterval1, ChargeCode_PulsesOnAnswer1, ChargeCode_EndTime2, ChargeCode_PulseInterval2, ChargeCode_PulsesOnAnswer2, ChargeCode_EndTime3, ChargeCode_PulseInterval3, ChargeCode_PulsesOnAnswer3, ChargeCode_EndTime4, ChargeCode_PulseInterval4, ChargeCode_PulsesOnAnswer4; [\ChargeCode] Where, ▪ EndTime = Period (1 - 4) end time. ▪ PulseInterval = Period (1 - 4) pulse interval. ▪ PulsesOnAnswer = Period (1 - 4) pulses on answer. For example: ChargeCode 1 = 7,30,1,14,20,2,20,15,1,0,60,1; ChargeCode 2 = 5,60,1,14,20,1,0,60,1; ChargeCode 3 = 0,60,1; ChargeCode 0 = 6, 3, 1, 12, 2, 1, 18, 5, 2, 0, 2, 1; **Notes:** ▪ The parameter can include up to 25 indices (i.e., up to 25 different metering rules can be defined). ▪ This parameter is applicable only to FXS interfaces. ▪ To associate a charge code to an outgoing Tel-to-IP call, use the 'Outbound IP Routing Table'. ▪ To configure the Charge Codes table using the Web interface, refer to "Configuring the Charge Codes Table" on page 139. ▪ For an explanation on configuration using *ini* file table parameters, refer to "Configuring *ini* File Table Parameters" on page 222.

6.15 Telephone Keypad Sequence Parameters

The telephony keypad sequence parameters are described in the table below.

Table 6-58: Keypad Sequence Parameters

Parameter	Description
Prefix for External Line	
[Prefix2ExtLine]	Defines a string prefix (e.g., '9' dialed for an external line) that when dialed, the device plays a secondary dial tone (i.e., stutter tone) to the FXS line and then starts collecting the subsequently dialed digits from the FXS line. The valid range is a one-character string. The default is an empty string. **Notes:** ▪ You can enable the device to add this string as the prefix to the collected (and sent) digits, using the parameter AddPrefix2ExtLine. ▪ This parameter is applicable only to FXS interfaces.
[AddPrefix2ExtLine]	Determines whether the prefix string for accessing an external line (defined by the parameter Prefix2ExtLine) is added to the dialed number as the prefix and together sent to the IP destination (Tel-to-IP calls). ▪ **[0]** = Disable (default) ▪ **[1]** = Enable For example, if this parameter is enabled and the prefix string for the external line is defined as "9" (using the parameter Prefix2ExtLine) and the FXS user wants to make a call to destination "123", the device collects and sends all the dialed digits, including the prefix string, as "9123" to the IP destination number. **Note:** This parameter is applicable only to FXS interfaces.
Hook Flash Parameters	
Web: Flash Keys Sequence Style **[FlashKeysSequenceStyle]**	Hook flash keys sequence style. ▪ **[0]** 0 = Flash hook (default) - only the phone's Flash button is used, according to the following scenarios: ✓ During an existing call, if the user presses Flash, the call is put on hold; a dial tone is heard and the user is able to initiate a second call. Once the second call is established, on-hooking transfers the first (held) call to the second call. ✓ During an existing call, if a call comes in (call waiting), pressing Flash places the active call on hold and answers the waiting call; pressing Flash again toggles between these two calls. ▪ **[1]** 1 = Sequence of Flash hook + digit: ✓ Flash + 1: holds a call or toggles between two existing calls ✓ Flash + 2: makes a call transfer. ✓ Flash + 3: makes a three-way conference call (if the

Parameter	Description
	Three-Way Conference feature is enabled, i.e., the parameter Enable3WayConference is set to 1 and the parameter 3WayConferenceMode is set to 2).
Web: Flash Keys Sequence Timeout **[FlashKeysSequenceTimeout]**	Flash keys sequence timeout - the time (in msec) that the device waits for digits after the user presses the Flash button (Flash Hook + Digit mode - when the parameter FlashKeysSequenceStyle is set to 1). The valid range is 100 to 5,000. The default is 2,000.
Keypad Feature - Call Forward Parameters	
Web: Unconditional EMS: Call Forward Unconditional **[KeyCFUnCond]**	Keypad sequence that activates the immediate call forward option.
Web: No Answer EMS: Call Forward No Answer **[KeyCFNoAnswer]**	Keypad sequence that activates the forward on no answer option.
Web: On Busy EMS: Call Forward Busy **[KeyCFBusy]**	Keypad sequence that activates the forward on busy option.
Web: On Busy or No Answer EMS: CF Busy Or No Answer **[KeyCFBusyOrNoAnswer]**	Keypad sequence that activates the forward on 'busy or no answer' option.
Web: Do Not Disturb EMS: CF Do Not Disturb **[KeyCFDoNotDisturb]**	Keypad sequence that activates the Do Not Disturb option (immediately reject incoming calls).
To activate the required forward method from the telephone: 1 Dial the user-defined sequence number on the keypad; a dial tone is heard. 2 Dial the telephone number to which the call is forwarded (terminate the number with #); a confirmation tone is heard.	
Web: Deactivate EMS: Call Forward Deactivation **[KeyCFDeact]**	Keypad sequence that deactivates any of the call forward options. After the sequence is pressed, a confirmation tone is heard.
Keypad Feature - Caller ID Restriction Parameters	
Web: Activate EMS: CLIR **[KeyCLIR]**	Keypad sequence that activates the restricted Caller ID option. After the sequence is pressed, a confirmation tone is heard.
Web: Deactivate EMS: CLIR Deactivation **[KeyCLIRDeact]**	Keypad sequence that deactivates the restricted Caller ID option. After the sequence is pressed, a confirmation tone is heard.
Keypad Feature - Hotline Parameters	
Web: Activate EMS: Hot Line **[KeyHotLine]**	Keypad sequence that activates the delayed hotline option. To activate the delayed hotline option from the telephone, perform the following: 1 Dial the user-defined sequence number on the keypad; a dial tone is heard. 2 Dial the telephone number to which the phone automatically dials after a configurable delay (terminate the number with #); a confirmation tone is heard.

Parameter	Description
Web: Deactivate EMS: Hot Line Deactivation **[KeyHotLineDeact]**	Keypad sequence that deactivates the delayed hotline option. After the sequence is pressed, a confirmation tone is heard.
Keypad Feature - Transfer Parameters	
Web: Blind EMS: Blind Transfer **[KeyBlindTransfer]**	Keypad sequence that activates blind transfer for Tel-to-IP calls. There are two possible scenarios: ▪ Option 1: After this sequence is dialed, the current call is put on hold (using Re-INVITE), a dial tone is played to the phone and then phone number collection starts. ▪ Option 2: A Hook-Flash is pressed, the current call is put on hold, a dial tone is played to the phone, and then digit collection starts. After this sequence is identified, the device continues the collection of the destination phone number. For both options, after the phone number is collected, it's sent to the transferee in a SIP REFER request (without a Replaces header). The call is then terminated and a confirmation tone is played to the phone. If the phone number collection fails due to a mismatch, a reorder tone is played to the phone. **Notes:** ▪ This parameter is applicable to FXO and FXS interfaces (but for FXO the Web interface does not display this parameter). ▪ It is possible to configure whether the KeyBlindTransfer code is added as a prefix to the dialed destination number, by using the parameter KeyBlindTransferAddPrefix.
Keypad Feature - Call Waiting Parameters	
Web: Activate EMS: Keypad Features CW **[KeyCallWaiting]**	Keypad sequence that activates the Call Waiting option. After the sequence is pressed, a confirmation tone is heard.
Web: Deactivate EMS: Keypad Features CW Deact **[KeyCallWaitingDeact]**	Keypad sequence that deactivates the Call Waiting option. After the sequence is pressed, a confirmation tone is heard.
Keypad Feature - Reject Anonymous Call Parameters	
Web: Activate EMS: Reject Anonymous Call **[KeyRejectAnonymousCall]**	Keypad sequence that activates the reject anonymous call option, whereby the device rejects incoming anonymous calls. After the sequence is pressed, a confirmation tone is heard.
Web: Deactivate EMS: Reject Anonymous Call Deact **[KeyRejectAnonymousCallDeact]**	Keypad sequence that de-activates the reject anonymous call option. After the sequence is pressed, a confirmation tone is heard.

Parameter	Description
[RejectAnonymousCallPerPort]	This *ini* file table parameter determines whether the device rejects incoming anonymous calls on FXS interfaces. The format of this parameter is as follows: [RejectAnonymousCallPerPort] FORMAT RejectAnonymousCallPerPort_Index = RejectAnonymousCallPerPort_Enable, RejectAnonymousCallPerPort_Port, RejectAnonymousCallPerPort_Module; [\RejectAnonymousCallPerPort] Where, ▪ Enable = accept **[0]** (default) or reject **[1]** incoming anonymous calls. ▪ Port = Port number. ▪ Module = Module number. For example: [RejectAnonymousCallPerPort] RejectAnonymousCallPerPort 0 = 0,1,1; RejectAnonymousCallPerPort 1 = 1,2,1; [\RejectAnonymousCallPerPort] If enabled, when a device's FXS interface receives an anonymous call, it responds with a 433 (Anonymity Disallowed) SIP response. **Notes:** ▪ This parameter is applicable only to FXS interfaces. ▪ This parameter is per FXS port. ▪ For an explanation on using *ini* file table parameters, refer to "Configuring *ini* File Table Parameters" on page 222.

6.16 General FXO Parameters

The general FXO parameters are described in the table below.

Table 6-59: General FXO Parameters

Parameter	Description
Web: FXO Coefficient Type EMS: Country Coefficients **[CountryCoefficients]**	Determines the FXO line characteristics (AC and DC) according to USA or TBR21 standard. ▪ **[66]** Europe = TBR21 ▪ **[70]** USA = United States (default) **Note:** For this parameter to take effect, a device reset is required.
[FXONumberOfRings]	Defines the number of rings before the device's FXO interface answers a call by seizing the line. The valid range is 0 to 10. The default is 0. When set to 0, the FXO seizes the line after one ring. When set to 1, the FXO seizes the line after two rings. **Notes:** ▪ This parameter is applicable only if automatic dialing is not used. ▪ If caller ID is enabled and if the number of rings defined by the parameter RingsBeforeCallerID is greater than the number of rings defined by this parameter, the greater value is used.
Web/EMS: Dialing Mode **[IsTwoStageDial]**	Determines the dialing mode for IP-to-Tel (FXO) calls. ▪ **[0]** One Stage = One-stage dialing. ▪ **[1]** Two Stages = Two-stage dialing (default). If two-stage dialing is enabled, the device seizes one of the PSTN/PBX lines without performing any dialing, connects the remote IP user to the PSTN/PBX, and all further signaling (dialing and Call Progress Tones) is performed directly with the PBX without the device's intervention. If one-stage dialing is enabled, the device seizes one of the available lines (according to the parameter ChannelSelectMode) and dials the destination phone number received in the INVITE message. Use the parameter IsWaitForDialTone to specify whether the dialing must start after detection of the dial tone or immediately after seizing the line. **Note:** This parameter is applicable only to FXO interfaces.
Web/EMS: Waiting For Dial Tone **[IsWaitForDialTone]**	Determines whether the device waits for a dial tone before dialing the phone number for IP-to-Tel (FXO) calls. ▪ **[0]** No = Don't wait for dial tone. ▪ **[1]** Yes = Wait for dial tone (default). When one-stage dialing and this parameter are enabled, the device dials the phone number (to the PSTN/PBX line) only after it detects a dial tone. If this parameter is disabled, the device immediately dials the phone number after seizing the PSTN/PBX line without 'listening' for a dial tone. **Notes:**

Parameter	Description
	▪ The correct dial tone parameters must be configured in the CPT file. ▪ The device may take 1 to 3 seconds to detect a dial tone (according to the dial tone configuration in the CPT file). If the dial tone is not detected within 6 seconds, the device releases the call and sends a SIP 500 "Server Internal Error" response. ▪ This parameter is applicable only to FXO interfaces.
Web: Time to Wait before Dialing [msec] EMS: Time Before Dial **[WaitForDialTime]**	For digital interfaces: Determines the delay after hook-flash is generated and until dialing begins. Applies to call transfer (i.e., the parameter TrunkTransferMode is set to 3) on CAS protocols. For Analog interfaces: Determines the delay before the device starts dialing on the FXO line in the following scenarios: ▪ The delay between the time the line is seized and dialing begins during the establishment of an IP-to-Tel call. **Note:** Applicable only for one-stage dialing when the parameter IsWaitForDialTone is disabled. ▪ The delay between detection of a Wink and the start of dialing during the establishment of an IP-to-Tel call (for DID lines, EnableDIDWink is set to 1). ▪ For call transfer - the delay after hook-flash is generated and dialing begins. The valid range (in milliseconds) is 0 to 20,000 (i.e., 20 seconds). The default value is 1,000 (i.e., 1 second).
Web: Ring Detection Timeout [sec] EMS: Timeout Between Rings **[FXOBetweenRingTime]**	Defines the timeout (in seconds) for detecting the second ring after the first detected ring. If automatic dialing is not used and Caller ID is enabled, the device seizes the line after detection of the second ring signal (allowing detection of caller ID sent between the first and the second rings). If the second ring signal is not received within this timeout, the device doesn't initiate a call to IP. If automatic dialing is used, the deviceinitiates a call to IP when the ringing signal is detected. The FXO line is seized only if the remote IP party answers the call. If the remote party doesn't answer the call and the second ring signal is not received within this timeout, the device releases the IP call. This parameter is typically set to between 5 and 8. The default is 8. **Notes:** ▪ This parameter is applicable only to FXO interfaces (for Tel-to-IP calls). ▪ This timeout is calculated from the end of the ring until the start of the next ring. For example, if the ring cycle is two seconds on and four seconds off, the timeout value should be configured to five seconds (i.e., greater than the off time, e.g., four).
Web: Rings before Detecting Caller ID EMS: Rings Before Caller ID **[RingsBeforeCallerID]**	Determines the number of rings before the device starts detecting Caller ID. ▪ **[0]** 0 = Before first ring. ▪ **[1]** 1 = After first ring (default). ▪ **[2]** 2 = After second ring. **Note:** This parameter is applicable only to FXO interfaces.

Parameter	Description
Web/EMS: Guard Time Between Calls **[GuardTimeBetweenCalls]**	Defines the time interval (in seconds) after a call has ended and a new call can be accepted for IP-to-Tel (FXO) calls. The valid range is 0 to 10. The default value is 1. **Notes:** ▪ Occasionally, after a call ends and on-hook is applied, a delay is required before placing a new call (and performing off-hook). This is necessary to prevent incorrect hook-flash detection or other glare phenomena. ▪ This parameter is applicable only to FXO interfaces.

6.17 FXS Parameters

The general FXS parameters are described in the table below.

Table 6-60: General FXS Parameters

Parameter	Description
Web: FXS Coefficient Type EMS: Country Coefficients **[FXSCountryCoefficients]**	Determines the FXS line characteristics (AC and DC) according to USA or Europe (TBR21) standards. ▪ **[66]** Europe = TBR21 ▪ **[70]** USA = United States (default) **Note:** For this parameter to take effect, a device reset is required.

6.18 Trunk Groups, Number Manipulation and Routing Parameters

This subsection describes the device's number manipulation and routing parameters.

6.18.1 Trunk Groups and Routing Parameters

The routing parameters are described in the table below.

Table 6-61: Routing Parameters

Parameter	Description
Web: Trunk Group Table **EMS: SIP Endpoints > Phones**	
[TrunkGroup]	This *ini* file table parameter is used to define and activate the device's endpoints/Trunk channels, by defining telephone numbers and assigning them to Trunk Groups. The format of this parameter is shown below: [TrunkGroup] FORMAT TrunkGroup_Index = TrunkGroup_TrunkGroupNum, TrunkGroup_FirstTrunkId, TrunkGroup_FirstBChannel, TrunkGroup_LastBChannel, TrunkGroup_FirstPhoneNumber, TrunkGroup_ProfileId, TrunkGroup_LastTrunkId, TrunkGroup_Module; [\TrunkGroup] For example: ▪ The configuration below assigns Trunk 1 (channels 1 to 30) of Module 1 to Trunk Group ID 2: TrunkGroup 0 = 2, 0, 1, 30, 50000, 0, 0, 1; ▪ The configuration below assigns BRI channels 1 through 4 of Module 2 to Trunk Group ID 2 with phone numbers 208 to 211: TrunkGroup 1 = 2, 0, 1, 4, 208, 0, 0 ,2; **Notes:** ▪ The first entry in this table starts at index 0. ▪ Trunk Group ID 1 is depicted as 0 in the table. ▪ This parameter can appear up to four times per module. ▪ For configuring this table in the Web interface, refer to Configuring the Trunk Group Table on page 102. ▪ For a description of *ini* file table parameters, refer to "Configuring *ini* File Table Parameters" on page 222.
Web: Trunk Group Settings **EMS: SIP Routing > Hunt Group**	
[TrunkGroupSettings]	This *ini* file table parameter defines rules for channel allocation per Trunk Group. If no rule exists, the rule defined by the global parameter ChannelSelectMode takes effect. The format of this parameter is as follows: [TrunkGroupSettings] FORMAT TrunkGroupSettings_Index =

Parameter	Description
	TrunkGroupSettings_TrunkGroupId, TrunkGroupSettings_ChannelSelectMode, TrunkGroupSettings_RegistrationMode, TrunkGroupSettings_GatewayName,TrunkGroupSettings_ContactUser, TrunkGroupSettings_ServingIPGroup, TrunkGroupSettings_MWIInterrogationType; [\TrunkGroupSettings] Where, ▪ MWIInterrogationType = defines QSIG MWI to IP interworking for interrogating MWI supplementary services: ✓ **[255]** Not Configured ✓ **[0]** None = disables the feature. ✓ **[1]** Use Activate Only = don't send any MWI Interrogation messages and only "passively" respond to MWI Activate requests from the PBX. ✓ **[2]** Result Not Used = send MWI Interrogation message, but don't use its result. Instead, wait for MWI Activate requests from the PBX. ✓ **[3]** Use Result = send MWI Interrogation messages, use its results, and use the MWI Activate requests. MWI Activate requests are interworked to SIP NOTIFY MWI messages. The SIP NOTIFY messages are sent to the IP Group defined by the NotificationIPGroupID parameter. For example: [TrunkGroupSettings] TrunkGroupSettings 0 = 1, 0, 5, branch-hq, user, 1, 255; TrunkGroupSettings 1 = 2, 1, 0, localname, user1, 2, 255; [\TrunkGroupSettings] **Notes:** ▪ This parameter can include up to 120 indices. ▪ For configuring Trunk Group Settings using the Web interface, refer to "Configuring Trunk Group Settings" on page 104. ▪ For a description on using *ini* file table parameters, refer to to "Configuring *ini* File Table Parameters" on page 222.
Web: Channel Select Mode EMS: Channel Selection Mode **[ChannelSelectMode]**	Method for allocating incoming IP-to-Tel calls to a channel. ▪ **[0]** By Dest Phone Number = Selects the device's channel according to the called number(default.) ▪ **[1]** Cyclic Ascending = Selects the next available channel in an ascending cyclic order. Always selects the next higher channel number in the Trunk Group. When the device reaches the highest channel number in the Trunk Group, it selects the lowest channel number in the Trunk Group and then starts ascending again. ▪ **[2]** Ascending = Selects the lowest available channel. It always starts at the lowest channel number in the Trunk Group and if that channel is unavailable, selects the next higher channel. ▪ **[3]** Cyclic Descending = Selects the next available channel in descending cyclic order. It always selects the next lower

Parameter	Description
	channel number in the Trunk Group. When the device reaches the lowest channel number in the Trunk Group, it selects the highest channel number in the Trunk Group and then starts descending again. ▪ **[4]** Descending = Selects the highest available channel. It always starts at the highest channel number in the Trunk Group and if that channel is unavailable, selects the next lower channel. ▪ **[5]** Dest Number + Cyclic Ascending = The device first selects the channel according to the called number. If the called number isn't found, it then selects the next available channel in ascending cyclic order. Note that if the called number is found but the port associated with this number is busy, the call is released. ▪ **[6]** By Source Phone Number = The device selects the channel according to the calling number. ▪ **[7]** Trunk Cyclic Ascending = The device selects the channel from the first channel of the next trunk (adjacent to the trunk from which the previous channel was allocated). This option is applicable only to digital interfaces. ▪ **[8]** Trunk & Channel Cyclic Ascending = The device implements the Trunk Cyclic Ascending and Cyclic Ascending methods to select the channel. This method selects the next physical trunk (pertaining to the Trunk Group) and then selects the B-channel of this trunk according to the cyclic ascending method (i.e., selects the channel after the last allocated channel). This option is applicable only to digital interfaces. For example, if the Trunk Group includes two physical trunks, 0 and 1: ✓ For the first incoming call, the first channel of Trunk 0 is allocated. ✓ For the second incoming call, the first channel of Trunk 1 is allocated. ✓ For the third incoming call, the second channel of Trunk 0 is allocated. **Notes:** ▪ For defining the channel select mode per Trunk Group, refer to "Configuring Trunk Group Settings" on page 104. ▪ The logical (for digital interfaces) phone numbers of the device's B-channels are defined by the TrunkGroup parameter.
Web: Default Destination Number **[DefaultNumber]**	Defines the default destination phone number, which is used if the received message doesn't contain a called party number and no phone number is configured in the 'Trunk Group Table' (refer to “Configuring the Trunk Group Table” on page 102). This parameter is used as a starting number for the list of channels comprising all the device's Trunk Groups. The default value is 1000.

Parameter	Description
Web: Source IP Address Input **[SourceIPAddressInput]**	Determines the IP address that the device uses to determine the source of incoming INVITE messages for IP-to-Tel routing. ▪ **[-1]** = Auto Decision - if the IP-to-IP feature is enabled, this parameter is automatically set to Layer 3 Source IP. If the IP-to-IP feature is disabled, this parameter is automatically set to SIP Contact Header (1). (default) ▪ **[0]** SIP Contact Header = The IP address in the Contact header of the incoming INVITE message is used. ▪ **[1]** Layer 3 Source IP = The actual IP address (Layer 3) from where the SIP packet was received is used.
Web: Use Source Number As Display Name EMS: Display Name **[UseSourceNumberAsDisplayName]**	Determines the use of Tel Source Number and Display Name for Tel-to-IP calls. ▪ **[0]** No = If a Tel Display Name is received, the Tel Source Number is used as the IP Source Number and the Tel Display Name is used as the IP Display Name. If no Display Name is received from the Tel side, the IP Display Name remains empty (default). ▪ **[1]** Yes = If a Tel Display Name is received, the Tel Source Number is used as the IP Source Number and the Tel Display Name is used as the IP Display Name. If no Display Name is received from the Tel side, the Tel Source Number is used as the IP Source Number and also as the IP Display Name. ▪ **[2]** Overwrite = The Tel Source Number is used as the IP Source Number and also as the IP Display Name (even if the received Tel Display Name is not empty).
Web/EMS: Use Display Name as Source Number **[UseDisplayNameAsSourceNumber]**	Determines the use of Source Number and Display Name for IP-to-Tel calls. ▪ **[0]** No = If IP Display Name is received, the IP Source Number is used as the Tel Source Number and the IP Display Name is used as the Tel Display Name. If no Display Name is received from IP, the Tel Display Name remains empty (default). ▪ **[1]** Yes = If an IP Display Name is received, it is used as the Tel Source Number and also as the Tel Display Name, and Presentation is set to Allowed (0). If no Display Name is received from IP, the IP Source Number is used as the Tel Source Number and Presentation is set to Restricted (1). For example: When 'From: 100 <sip:200@201.202.203.204>' is received, the outgoing Source Number and Display Name are set to '100' and the Presentation is set to Allowed (0). When 'From: <sip:100@101.102.103.104>' is received, the outgoing Source Number is set to '100' and the Presentation is set to Restricted (1).
Web: Use Routing Table for Host Names and Profiles EMS: Use Routing Table For Host Names **[AlwaysUseRouteTable]**	Determines whether to use the device's routing table to obtain the URI host name and optionally, an IP profile (per call) even if a Proxy server is used. ▪ **[0]** Disable = Don't use internal routing table (default). ▪ **[1]** Enable = Use the 'Outbound IP Routing Table'.

Parameter	Description
	Notes: ▪ This parameter appears only if the 'Use Default Proxy' parameter is enabled. ▪ The domain name is used instead of a Proxy name or IP address in the INVITE SIP URI.
Web/EMS: Tel to IP Routing Mode **[RouteModeTel2IP]**	For a description of this parameter, refer to "Configuring the Outbound IP Routing Table" on page 158.
Web: Outbound IP Routing Table **EMS: SIP Routing > Tel to IP**	
[Prefix]	This *ini* file table parameter configures the 'Outbound IP Routing Table' for routing Tel-to-IP calls and IP-to-IP calls. The format of this parameter is as follows: [PREFIX] FORMAT PREFIX_Index = PREFIX_DestinationPrefix, PREFIX_DestAddress, PREFIX_SourcePrefix, PREFIX_ProfileId, PREFIX_MeteringCode, PREFIX_DestPort, PREFIX_SrcIPGroupID, PREFIX_DestHostPrefix, PREFIX_DestIPGroupID, PREFIX_SrcHostPrefix, PREFIX_TransportType, PREFIX_SrcTrunkGroupID; [\PREFIX] For example: PREFIX 0 = *, quest, *, 0, 255, $$, -1, , 1, , -1, -1; PREFIX 1 = 20, 10.33.37.77, *, 0, 255, $$, -1, , 2, , 0, -1; PREFIX 2 = 30, 10.33.37.79, *, 1, 255, $$, -1, , -1, , 2, -1; **Notes:** ▪ This parameter can include up to 200 indices. ▪ For a detailed description of the table's parameters and for configuring this table using the Web interface, refer to "Configuring the Outbound IP Routing Table" on page 158. ▪ For a description on using *ini* file table parameters, refer to "Configuring *ini* File Table Parameters" on page 222.
Web: Inbound IP Routing Table **EMS: SIP Routing > IP to Hunt**	
[PSTNPrefix]	This *ini* file table parameter configures the routing of IP calls to Trunk Groups (or inbound IP Groups). The format of this parameter is as follows: [PSTNPrefix] FORMAT PstnPrefix_Index = PstnPrefix_DestPrefix, PstnPrefix_TrunkGroupId, PstnPrefix_SourcePrefix, PstnPrefix_SourceAddress, PstnPrefix_ProfileId, PstnPrefix_SrcIPGroupID, PstnPrefix_DestHostPrefix, PstnPrefix_SrcHostPrefix; [\PSTNPrefix] For example: PstnPrefix 0 = 100, 1, 200, *, 0, 2, , ; PstnPrefix 1 = *, 2, *, , 1, 3, acl, joe; **Notes:** ▪ This parameter can include up to 24 indices.

Parameter	Description
	▪ For a description of the table's parameters, refer to the corresponding Web parameters in "Configuring the Inbound IP Routing Table" on page 163. ▪ To support the In-Call Alternative Routing feature, you can use two entries that support the same call but assigned with a different Trunk Group. The second entry functions as an alternative route if the first rule fails as a result of one of the release reasons configured in the AltRouteCauseIP2Tel table. ▪ Selection of Trunk Groups (for IP-to-Tel calls) is according to destination number, source number,and source IP address. ▪ The source IP address (SourceAddress) can include the 'x' wildcard to represent single digits. For example: 10.8.8.xx represents all IP addresses between 10.8.8.10 and 10.8.8.99. ▪ The source IP address (SourceAddress) can include the asterisk ('*') wildcard to represent any number between 0 and 255. For example, 10.8.8.* represents all addresses between 10.8.8.0 and 10.8.8.255. ▪ If the source IP address (SourceAddress) includes an FQDN, DNS resolution is performed according to the parameter DNSQueryType. ▪ For available notations for depicting a range of multiple numbers, refer to "Dialing Plan Notation for Routing and Manipulation" on page 485. ▪ For a description on using *ini* file table parameters, refer to "Configuring *ini* File Table Parameters" on page 222.
Web/EMS: IP to Tel Routing Mode **[RouteModeIP2Tel]**	Determines whether to route IP calls to the Trunk Group (or IP Group) before or after manipulation of the destination number (configured in "Configuring the Number Manipulation Tables" on page 143). ▪ **[0]** Route calls before manipulation = Calls are routed before the number manipulation rules are applied (default). ▪ **[1]** Route calls after manipulation = Calls are routed after the number manipulation rules are applied.
Web: IP Security EMS: Secure Call From IP **[SecureCallsFromIP]**	Determines whether the device accepts SIP calls only from configured SIP Proxies or IP addresses defined in the 'Outbound IP Routing Table' (refer to "Configuring the Outbound IP Routing Table" on page 158). This is useful in preventing unwanted SIP calls, SIP messages, and/or VoIP spam. ▪ **[0]** Disable = The device accepts all SIP calls (default). ▪ **[1]** Enable = The device accepts SIP calls only from IP addresses defined in the 'Outbound IP Routing Table' and rejects all other calls. **Notes:** ▪ When using Proxies or Proxy Sets, it is unnecessary to configure the Proxy IP addresses in the routing table. The device allows SIP calls received from the Proxy IP addresses even if these addresses are not configured in the routing table.

Parameter	Description
	▪ This feature is supported only for numerical IP addresses in the 'Outbound IP Routing Table'.
Web/EMS: Filter Calls to IP **[FilterCalls2IP]**	Enables filtering of Tel-to-IP calls when a Proxy is used (i.e., IsProxyUsed parameter is set to 1 - refer to "Configuring Proxy and Registration Parameters" on page 122). ▪ **[0]** Don't Filter = device doesn't filter calls when using a Proxy (default). ▪ **[1]** Filter = Filtering is enabled. When this parameter is enabled and a Proxy is used, the device first checks the 'Outbound IP Routing Table' before making a call through the Proxy. If the number is not allowed (i.e., number isn't listed in the table or a call restriction routing rule of IP address 0.0.0.0 is applied), the call is released. **Note:** When no Proxy is used, this parameter must be disabled and filtering is according to the 'Outbound IP Routing Table'.
[IP2TelTaggingDestDialPlanIndex]	Determines the Dial Plan index in the external Dial Plan file (*.dat) in which string labels ("tags") are defined for tagging incoming IP-to-Tel calls. The special “tag” is added as a prefix to the called party number, and then the 'Inbound IP Routing Table' uses this “tag” instead of the original prefix. Manipulation is then performed (after routing) in the Manipulation table which strips the “tag” characters before sending the call to the endpoint. The valid values are 0 to 7, where 0 denotes PLAN1, 1 denotes PLAN2, and so on. The default is -1 (i.e., no dial plan file used). The routing label can be up to 9 (text) characters. **Notes:** ▪ This parameter is applicable only to digital interfaces. ▪ The routing must be configured to be performed before manipulation. ▪ For a detailed description of this feature, refer to Dial Plan Prefix Tags for IP-to-Tel Routing on page 490.
[EnableETSIDiversion]	Defines the method in which the Redirect Number is sent to the Tel side. ▪ **[0]** = Q.931 Redirecting Number Information Element (IE) (default) ▪ **[1]** = ETSI DivertingLegInformation2 in a Facility IE
Web: Add CIC **[AddCicAsPrefix]**	Determines whether to add the Carrier Identification Code (CIC) as a prefix to the destination phone number for IP-to-Tel calls. ▪ **[0]** No (default) ▪ **[1]** Yes When this parameter is enabled, the cic parameter in the incoming SIP INVITE can be used for IP-to-Tel routing decisions. It routes the call to the appropriate Trunk Group based on this parameter's value. The SIP cic parameter enables the transmission of the cic parameter from the SIP network to the ISDN. The cic parameter is a three- or four-digit code used in routing tables to identify the network that serves the remote user when a call is routed over

Parameter	Description
	many different networks. The cic parameter is carried in the SIP INVITE and maps to the ISDN Transit Network Selection Information Element (TNS IE) in the outgoing ISDN Setup message (if the EnableCIC parameter is set to 1). The TNS IE identifies the requested transportation networks and allows different providers equal access support, based on customer choice. For example, as a result of receiving the below INVITE, the destination number after number manipulation is cic+167895550001: INVITE sip:5550001;cic=+16789@172.18.202.60:5060;user=phone SIP/2.0 **Note:** After the cic prefix is added, the 'Inbound IP Routing Table' can be used to route this call to a specific Trunk Group. The Destination Number IP to Tel Manipulation table must be used to remove this prefix before placing the call to the ISDN.

6.18.2 Alternative Routing Parameters

The alternative routing parameters are described in the table below.

Table 6-62: Alternative Routing Parameters

Parameter	Description
Web/EMS: Redundant Routing Mode **[RedundantRoutingMode]**	Determines the type of redundant routing mechanism when a call can't be completed using the main route. ▪ **[0]** Disable = No redundant routing is used. If the call can't be completed using the main route (using the active Proxy or the first matching rule in the Routing table), the call is disconnected. ▪ **[1]** Routing Table = Internal routing table is used to locate a redundant route (default). ▪ **[2]** Proxy = Proxy list is used to locate a redundant route. **Note:** To implement the Redundant Routing Mode mechansim, you first need to configure the parameter AltRouteCauseTEL2IP (Reasons for Alternative Routing table).
Web: Enable Alt Routing Tel to IP EMS: Enable Alternative Routing **[AltRoutingTel2IPEnable]**	Enables the Alternative Routing feature for Tel-to-IP calls. ▪ **[0]** Disable = Disables the Alternative Routing feature (default). ▪ **[1]** Enable = Enables the Alternative Routing feature. ▪ **[2]** Status Only = The Alternative Routing feature is disabled, but read-only information on the QoS of the destination IP addresses is provided. For information on the Alternative Routing feature, refer to "Configuring Alternative Routing (Based on Connectivity and QoS)" on page 554.

Parameter	Description
Web: Alt Routing Tel to IP Mode EMS: Alternative Routing Mode **[AltRoutingTel2IPMode]**	Determines the event(s) reason for triggering Alternative Routing. ▪ **[0]** None = Alternative routing is not used. ▪ **[1]** Connectivity = Alternative routing is performed if a ping to the initial destination fails. ▪ **[2]** QoS = Alternative routing is performed if poor QoS is detected. ▪ **[3]** Both = Alternative routing is performed if either ping to initial destination fails, poor QoS is detected, or the DNS host name is not resolved (default). **Notes:** ▪ QoS is quantified according to delay and packet loss calculated according to previous calls. QoS statistics are reset if no new data is received within two minutes. For information on the Alternative Routing feature, refer to "Configuring Alternative Routing (Based on Connectivity and QoS)" on page 554. ▪ To receive quality information (displayed in the 'Quality Status' and 'Quality Info.' fields in "Viewing IP Connectivity" on page 219) per destination, this parameter must be set to 2 or 3.
Web: Alt Routing Tel to IP Connectivity Method EMS: Alternative Routing Telephone to IP Connection Method **[AltRoutingTel2IPConnMethod]**	Determines the method used by the device for periodically querying the connectivity status of a destination IP address. ▪ **[0]** ICMP Ping (default) = Internet Control Message Protocol (ICMP) ping messages. ▪ **[1]** SIP OPTIONS = The remote destination is considered offline if the latest OPTIONS transaction timed out. Any response to an OPTIONS request, even if indicating an error, brings the connectivity status to online.
[EnableAltMapTel2IP]	Enables different Tel-to-IP destination number manipulation rules per routing rule when several (up to three) Tel-to-IP routing rules are defined and if alternative routing using release causes is used. For example, if an INVITE message for a Tel-to-IP call is returned with a SIP 404 Not Found response, the call can be re-sent to a different destination number (as defined using the parameter NumberMapTel2IP). ▪ **[0]** = Disable (default) ▪ **[1]** = Enable
Web: Alt Routing Tel to IP Keep Alive Time EMS: Alternative Routing Keep Alive Time **[AltRoutingTel2IPKeepAliveTime]**	Defines the time interval (in seconds) between SIP OPTIONS Keep-Alive messages used for the IP Connectivity application. The valid range is 5 to 2,000,000. The default value is 60.

Parameter	Description
Web/EMS: Alternative Routing Tone Duration **[ms]** **[AltRoutingToneDuration]**	Determines the duration (in milliseconds) for which the device plays a tone to the endpoint on each Alternative Routing attempt. When the device finishes playing the tone, a new SIP INVITE message is sent to the new destination. The tone played is the Call Forward Tone (Tone Type #25 in the CPT file). The valid range is 0 to 20,000. The default is 0 (i.e., no tone is played).
Web: Max Allowed Packet Loss for Alt Routing **[%]** **[IPConnQoSMaxAllowedPL]**	Packet loss in percentage at which the IP connection is considered a failure and Alternative Routing mechanism is activated. The default value is 20%.
Web: Max Allowed Delay for Alt Routing **[msec]** **[IPConnQoSMaxAllowedDelay]**	Transmission delay (in msec) at which the IP connection is considered a failure and the Alternative Routing mechanism is activated. The range is 100 to 10,000. The default value is 250.
Web: Reasons for Alternative Tel-to-IP Routing Table **EMS: Alt Route Cause Tel to IP**	
[AltRouteCauseTel2IP]	This *ini* file table parameter configures SIP call failure reason values received from the IP side. If an IP call is released as a result of one of these reasons, the device attempts to locate an alternative IP route (address) for the call in the 'Outbound IP Routing Table' (if a Proxy is not used) or used as a redundant Proxy (you need to set the parameter RedundantRoutingMode to 2). The release reason for Tel-to-IP calls is provided in SIP 4xx, 5xx, and 6xx response codes. The format of this parameter is as follows: [AltRouteCauseTel2IP] FORMAT AltRouteCauseTel2IP_Index = AltRouteCauseTel2IP_ReleaseCause; [\AltRouteCauseTel2IP] For example: AltRouteCauseTel2IP 0 = 486; (Busy Here) AltRouteCauseTel2IP 1 = 480; (Temporarily Unavailable) AltRouteCauseTel2IP 2 = 408; (No Response) **Notes:** ▪ This parameter can include up to 5 indices. ▪ The reasons for alternative routing for Tel-to-IP calls apply only when a Proxy is not used. ▪ When there is no response to an INVITE message (after INVITE retransmissions), the device issues an internal 408 'No Response' implicit release reason. ▪ The device sends the call to an alternative IP route only after the call has failed and the device has subsequently attempted twice to establish the call unsuccessfully. ▪ The device also plays a tone to the endpoint whenever an alternative route is used. This tone is played for a user-defined time (configured by the parameter AltRoutingToneDuration).

Parameter	Description
	▪ For an explanation on using *ini* file table parameters, refer to "Configuring *ini* File Table Parameters" on page 222
Web: Reasons for Alternative IP-to-Tel Routing Table **EMS: Alt Route Cause IP to Tel**	
[AltRouteCauseIP2Tel]	This *ini* file table parameter configures call failure reason values received from the PSTN side (in Q.931 presentation). If a call is released as a result of one of these reasons, the device attempts to locate an alternative Trunk Group for the call in the 'Inbound IP Routing Table'. The format of this parameter is as follows: [AltRouteCauseIP2Tel] FORMAT AltRouteCauseIP2Tel_Index = AltRouteCauseIP2Tel_ReleaseCause; [\AltRouteCauseIP2Tel] For example: AltRouteCauseIP2Tel 0 = 3 (No Route to Destination) AltRouteCauseIP2Tel 1 = 1 (Unallocated Number) AltRouteCauseIP2Tel 2 = 17 (Busy Here) **Notes:** ▪ This parameter can include up to 5 indices. ▪ If the device fails to establish a call to the PSTN because it has no available channels in a specific Trunk Group (e.g., all the channels are occupied, or the spans are disconnected or out-of-sync), it uses the Internal Release Cause '3' (No Route to Destination). This cause can be used in the AltRouteCauseIP2Tel table to define routing to an alternative Trunk Group. ▪ This table can be used for example, in scenarios where the destination is busy and the Release Reason #17 is issued or for other call releases that issue the default Release Reason (#3). ▪ The device also plays a tone to the endpoint whenever an alternative route is used. This tone is played for a user-defined time (configured by the parameter AltRoutingToneDuration). ▪ For an explanation on using *ini* file table parameters, refer to "Configuring *ini* File Table Parameters" on page 222.

Parameter	Description
Web: Forward On Busy Trunk Destination	
[ForwardOnBusyTrunkDest]	This *ini* file table parameter configures the Forward On Busy Trunk Destination table. This table allows you to define an alternative IP destination (IP address) per Trunk Group for IP-to Tel calls. The IP-to-Tel call is forwarded to this IP destination (using 3xx response) if the following exists: ▪ For digital PSTN interfaces: if a Trunk Group has no free channels (i.e., "busy" Trunk Group). ▪ For analog interfaces: if an FXO/FXS Trunk Group has no free channels. This feature can be used, for example, to forward the call to another FXO/FXS device. The device forwards calls using this table only if no alternative IP-to-Tel routing has been configured or alternative routing fails, and one of the following call forward reasons (included in the SIP Diversion header of 3xx messages) exist: ▪ "out-of-service" - all trunks are unavailable/disconnected (only for digital interfaces) ▪ "unavailable": ✓ All trunks are busy or unavailable (only for digital interfaces) ✓ All FXO/FXS lines pertaining to a Trunk Group are busy or unavailable The format of this parameter is as follows: [ForwardOnBusyTrunkDest] FORMAT ForwardOnBusyTrunkDest_Index = ForwardOnBusyTrunkDest_TrunkGroupId, ForwardOnBusyTrunkDest_ForwardDestination; [\ForwardOnBusyTrunkDest] For example, the below configuration forwards IP-to-Tel calls to destination IP address 10.13.4.12, port 5060 using transport protocol TCP, if Trunk Group ID 2 is busy: ForwardOnBusyTrunkDest 1 = 2, 10.13.4.12:5060;transport=tcp; **Notes:** ▪ The maximum number of indices (starting from 1) depends on the maximum number of Trunk Groups. ▪ For the destination, instead of a dotted-decimal IP address, FQDN can be used. In addition, the following syntax can be used: "host:port;transport=xxx"(i.e., IP address, port and transport type).

6.18.3 Number Manipulation Parameters

The number manipulation parameters are described in the table below.

Table 6-63: Number Manipulation Parameters

Parameter	Description
Web: Set Redirect number Screening Indicator to TEL EMS: Set IP To Tel Redirect Screening Indicator **[SetIp2TelRedirectScreeningInd]**	Defines the value of the Redirect Number screening indicator in ISDN Setup messages. ▪ **[-1]** Not Configured (default) ▪ **[0]** User Provided ▪ **[1]** User Passed ▪ **[2]** User Failed ▪ **[3]** Network Provided **Note:** This parameter is applicable only to digital PSTN interfaces (ISDN).
Web: Copy Destination Number to Redirect Number EMS: Copy Dest to Redirect Number **[CopyDest2RedirectNumber]**	Determines whether the device copies the received ISDN (digital interfaces) called number to the outgoing SIP Diversion header for Tel-to-IP calls (even if a Redirecting Number IE is not received in the ISDN Setup message, for digital interfaces). Therefore, the called number is used as a redirect number. Call redirection information is typically used for Unified Messaging and voice mail services to identify the recipient of a message. ▪ **[0]** Don't copy = Disable (default). ▪ **[1]** Copy after phone number manipulation = Copies the called number after manipulation. The device first performs Tel-to-IP destination phone number manipulation (i.e., on the SIP To header), and only then copies the manipulated called number to the SIP Diversion header for the Tel-to-IP call. Therefore, with this option, the called and redirected numbers are identical. ▪ **[2]** Copy before phone number manipulation = Copies the called number before manipulation. The device first copies the original called number to the SIP Diversion header and then performs Tel-to-IP destination phone number manipulation. Therefore, this allows you to have different numbers for the called (i.e., SIP To header) and redirected (i.e., SIP Diversion header) numbers. **Notes:** ▪ For digital interfaces: If the incoming ISDN-to-IP call includes a Redirect Number, this number is overridden by the new called number if this parameter is set to **[1]** or **[2].** ▪ This parameter can also be configured for IP Profiles (using the parameter IPProfile).

Parameter	Description
Web: Redirect Number IP -> Tel **EMS: Redirect Number Map IP to Tel**	
[RedirectNumberMapIp2Tel]	This *ini* file table parameter manipulates the redirect number for IP-to-Tel calls. This manipulates the value of the SIP Diversion, History-Info, or Resource-Priority headers (including the reason the call was redirected). The format of this parameter is as follows: [RedirectNumberMapIp2Tel] FORMAT RedirectNumberMapIp2Tel_Index = RedirectNumberMapIp2Tel_DestinationPrefix, RedirectNumberMapIp2Tel_RedirectPrefix, RedirectNumberMapIp2Tel_SourceAddress, RedirectNumberMapIp2Tel_NumberType, RedirectNumberMapIp2Tel_NumberPlan, RedirectNumberMapIp2Tel_RemoveFromLeft, RedirectNumberMapIp2Tel_RemoveFromRight, RedirectNumberMapIp2Tel_LeaveFromRight, RedirectNumberMapIp2Tel_Prefix2Add, RedirectNumberMapIp2Tel_Suffix2Add, RedirectNumberMapIp2Tel_IsPresentationRestricted; [\RedirectNumberMapIp2Tel] For example: RedirectNumberMapIp2Tel 1 = *, 88, *, 1, 1, 2, 0, 255, 9, , 255; **Notes:** ▪ This parameter table can include up to 20 indices (1-20). ▪ If the table's characteristics rule (i.e., DestinationPrefix, RedirectPrefix, and SourceAddress) matches the IP-to-Tel call, then the redirect number manipulation rule (defined by the other parameters) is applied to the call. ▪ The manipulation rules are executed in the following order: RemoveFromLeft, RemoveFromRight, LeaveFromRight, Prefix2Add, and then Suffix2Add. ▪ The RedirectPrefix parameter is used before any manipulation has been performed on it. ▪ The redirect manipulation is performed only after the parameter CopyDest2RedirectNumber.
Web: Redirect Number Tel -> IP **EMS: Redirect Number Map Tel to IP**	
[RedirectNumberMapTel2IP]	This *ini* file table parameter manipulates the redirect number for Tel-to-IP calls. The manipulated Redirect Number is sent in the SIP Diversion, History-Info, or Resource-Priority headers. The format of this parameter is as follows: [RedirectNumberMapTel2Ip] FORMAT RedirectNumberMapTel2Ip_Index = RedirectNumberMapTel2Ip_DestinationPrefix, RedirectNumberMapTel2Ip_RedirectPrefix, RedirectNumberMapTel2Ip_NumberType, RedirectNumberMapTel2Ip_NumberPlan,

Parameter	Description
	RedirectNumberMapTel2Ip_RemoveFromLeft, RedirectNumberMapTel2Ip_RemoveFromRight, RedirectNumberMapTel2Ip_LeaveFromRight, RedirectNumberMapTel2Ip_Prefix2Add, RedirectNumberMapTel2Ip_Suffix2Add, RedirectNumberMapTel2Ip_IsPresentationRestricted, RedirectNumberMapTel2Ip_SrcTrunkGroupID, RedirectNumberMapTel2Ip_SrcIPGroupID; [\RedirectNumberMapTel2Ip] For example: RedirectNumberMapTel2Ip 1 = *, 4, 255, 255, 0, 0, 255, , 972, 255, 1, 2; **Notes:** ▪ This parameter table can include up to 20 indices (1-20). ▪ If the table's matching characteristics rule (i.e., DestinationPrefix, RedirectPrefix, SrcTrunkGroupID, and SrcIPGroupID) is located for the Tel-to-IP call, then the redirect number manipulation rule (defined by the other parameters) is applied to the call. ▪ The manipulation rules are performed in the following order: RemoveFromLeft, RemoveFromRight, LeaveFromRight, Prefix2Add, and then Suffix2Add. ▪ The following parameters are applicable only to digital interfaces: NumberType, NumberPlan, and IsPresentationRestricted. ▪ The parameters NumberType and NumberPlan are applicable only to the SIP Resource-Priority header.
Phone-Context Parameters	
Web/EMS: Add Phone Context As Prefix **[AddPhoneContextAsPrefix]**	Determines whether the received Phone-Context parameter is added as a prefix to the outgoing ISDN Setup message with (for digital interfaces) Called and Calling numbers. ▪ **[0]** Disable = Disable (default). ▪ **[1]** Enable = Enable.
Web: Phone Context Table **EMS: SIP Manipulations > Phone Context**	
[PhoneContext]	This *ini* file table parameter defines the Phone Context table. This parameter maps NPI and TON to the SIP Phone-Context parameter. When a call is received from the ISDN/Tel, the NPI and TON are compared against the table and the corresponding Phone-Context value is used in the outgoing SIP INVITE message. The same mapping occurs when an INVITE with a Phone-Context attribute is received. The Phone-Context parameter appears in the standard SIP headers (Request-URI, To, From, Diversion) where a phone number is used. The format for this parameter is as follows: [PhoneContext] FORMAT PhoneContext_Index = PhoneContext_Npi, PhoneContext_Ton, PhoneContext_Context; [\PhoneContext]

Parameter	Description
	For example: PhoneContext 0 = 0,0,unknown.com PhoneContext 1 = 1,1,host.com PhoneContext 2 = 9,1,na.e164.host.com **Notes:** ▪ This parameter can include up to 20 indices. ▪ Several entries with the same NPI-TON or Phone-Context are allowed. In this scenario, a Tel-to-IP call uses the first match. ▪ Phone-Context '+' is unique in that it doesn't appear in the Request-URI as a Phone-Context parameter. Instead, it's added as a prefix to the phone number. The '+' isn't removed from the phone number in the IP-to-Tel direction. ▪ To configure the Phone Context table using the Web interface, refer to "Mapping NPI/TON to SIP Phone-Context" on page 153. ▪ For a description on using *ini* file table parameters, refer to "Configuring *ini* File Table Parameters" on page 222.
Web/EMS: Add Trunk Group ID as Prefix **[AddTrunkGroupAsPrefix]**	Determines whether the Trunk Group ID is added as a prefix to the destination phone number (i.e., called number) for Tel-to-IP calls. ▪ **[0]** No = Don't add Trunk Group ID as prefix (default). ▪ **[1]** Yes = Add Trunk Group ID as prefix to called number. **Notes:** ▪ This option can be used to define various routing rules. ▪ To use this feature, you must configure the Trunk Group IDs (refer to “Configuring the Trunk Group Table” on page 102).
Web: Add Trunk ID as Prefix EMS: Add Port ID As Prefix **[AddPortAsPrefix]**	Determines whether the slot number and port number/Trunk ID are added as a prefix to the called number for Tel-to-IP calls. ▪ **[0]** No = slot number and port number/Trunk ID not added as prefix (default). ▪ **[1]** Yes = slot number and port number/Trunk ID added as prefix. If enabled, the slot number (a single digit in the range of 1 to 6) and port number/Trunk ID (single digit in the range 1 to 8) are added as a prefix to the called (destination) phone number. For example, for the first trunk located in the first slot, the number "11" is added as the prefix. This option can be used to define various routing rules.
Web/EMS: Add Trunk Group ID as Prefix to Source **[AddTrunkGroupAsPrefixToSource]**	Determines whether the device adds the Trunk Group ID (from where the call originated) as the prefix to the calling number (i.e. source number). ▪ **[0]** No (default) ▪ **[1]** Yes

Parameter	Description
Web: Replace Empty Destination with B-channel Phone Number EMS: Replace Empty Dst With Port Number **[ReplaceEmptyDstWithPortNumber]**	Determines whether the internal channel number is used as the destination number if the called number is missing. ▪ **[0]** No (default) ▪ **[1]** Yes **Note:** This parameter is applicable only to Tel-to-IP calls and if the called number is missing.
[CopyDestOnEmptySource]	▪ **[0]** = Leave Source Number empty (default). ▪ **[1]** = If the Source Number of a Tel-to-IP call is empty, the Destination Number is copied to the Source Number.
Web: Add NPI and TON to Calling Number EMS: Add NPI And TON As Prefix To Calling Number **[AddNPIandTON2CallingNumber]**	Determines whether the Numbering Plan Indicator (NPI) and Type of Numbering (TON) are added to the Calling Number for Tel-to-IP calls. ▪ **[0]** No = Do not change the Calling Number (default). ▪ **[1]** Yes = Add NPI and TON to the Calling Number ISDN Tel-to-IP call. For example: After receiving a Calling Number of 555, NPI of 1, and TON of 3, the modified number becomes 13555. This number can later be used for manipulation and routing.
Web: Add NPI and TON to Called Number EMS: Add NPI And TON As Prefix To Called Number **[AddNPIandTON2CalledNumber]**	Determines whether NPI and TON are added to the Called Number for Tel-to-IP calls. ▪ **[0]** No = Do not change the Called Number (default). ▪ **[1]** Yes = Add NPI and TON to the Called Number of ISDN Tel-to-IP call. For example: After receiving a Called Number of 555, NPI of 1 and TON of 3, the modified number becomes 13555. This number can later be used for manipulation and routing.
Web: IP to Tel Remove Routing Table Prefix EMS: Remove Prefix **[RemovePrefix]**	Determines whether the device removes the prefix from the destination number for IP-to-Tel calls. ▪ **[0]** No = Don't remove prefix (default) ▪ **[1]** Yes = Remove the prefix (defined in the 'Inbound IP Routing Table' - refer to "Configuring the Inbound IP Routing Table" on page 163) from a telephone number for an IP-to-Tel call before forwarding it to Tel. For example: To route an incoming IP-to-Tel call with destination number 21100, the 'Inbound IP Routing Table' is scanned for a matching prefix. If such a prefix is found (e.g., 21), then before the call is routed to the corresponding Trunk Group, the prefix (21) is removed from the original number, and therefore, only 100 remains. **Notes:** ▪ This parameter is applicable only if number manipulation is performed after call routing for IP-to-Tel calls (i.e., RouteModeIP2Tel parameter is set to 0). ▪ Similar operation (of removing the prefix) is also achieved by using the usual number manipulation rules.

Parameter	Description
Web/EMS: Swap Redirect and Called Numbers **[SwapRedirectNumber]**	▪ **[0]** No = Don't change numbers (default). ▪ **[1]** Yes = Incoming ISDN call that includes a redirect number (sometimes referred to as 'original called number') uses the redirect number instead of the called number.
[SwapTel2IPCalled&CallingNumbers]	If enabled, the device swaps the calling and called numbers received from the Tel side (for Tel-to-IP calls). The SIP INVITE message contains the swapped numbers. ▪ **[0]** = Disabled (default) ▪ **[1]** = Swap calling and called numbers
Web/EMS: Add Prefix to Redirect Number **[Prefix2RedirectNumber]**	Defines a string prefix that is added to the Redirect number received from the Tel side. This prefix is added to the Redirect Number in the SIP Diversion header. The valid range is an 8-character string. The default is an empty string.
Web/EMS: Source Manipulation Mode **[SourceManipulationMode]**	Determines the SIP headers containing the source number after manipulation: ▪ **[0]** = The SIP From and P-Asserted-Identity headers contain the source number after manipulation (default). ▪ **[1]** = Only SIP From header contains the source number after manipulation, while the P-Asserted-Identity header contains the source number before manipulation.
Web: Add Number Plan and Type to RPI Header EMS: Add Ton 2 RPI **[AddTON2RPI]**	Determines whether the TON/PLAN parameters are included in the Remote-Party-ID (RPID) header. ▪ **[0]** No ▪ **[1]** Yes (default) If the Remote-Party-ID header is enabled (EnableRPIHeader = 1) and AddTON2RPI = 1, it's possible to configure the calling and called number type and number plan using the Number Manipulation tables for Tel-to-IP calls.
Web: Destination Phone Number Manipulation Table for Tel to IP Calls **EMS: SIP Manipulations > Destination Telcom to IPs**	
[NumberMapTel2IP]	This *ini* file table parameter manipulates the destination number of Tel-to-IP calls. The format of this parameter is as follows: [NumberMapTel2Ip] FORMAT NumberMapTel2Ip_Index = NumberMapTel2Ip_DestinationPrefix, NumberMapTel2Ip_SourcePrefix, NumberMapTel2Ip_SourceAddress, NumberMapTel2Ip_NumberType, NumberMapTel2Ip_NumberPlan, NumberMapTel2Ip_RemoveFromLeft, NumberMapTel2Ip_RemoveFromRight, NumberMapTel2Ip_LeaveFromRight, NumberMapTel2Ip_Prefix2Add, NumberMapTel2Ip_Suffix2Add,

Parameter	Description
	NumberMapTel2Ip_IsPresentationRestricted, NumberMapTel2Ip_SrcTrunkGroupID, NumberMapTel2Ip_SrcIPGroupID; [\NumberMapTel2Ip] For example: NumberMapTel2Ip 0 = 01,$$,*,0,0,2,$$,$$,971,$$,$$,$$,$$; NumberMapTel2Ip 1 = 10,10,*,255,255,3,0,5,100,$$,255,$$,$$; **Notes:** ▪ This table parameter can include up to 120 indices (0-119). ▪ The parameters SourceAddress and IsPresentationRestricted are not applicable. ▪ The parameters NumberType, NumberPlan, RemoveFromLeft, RemoveFromRight, Prefix2Add, Suffix2Add, and LeaveFromRight are applied if the called and calling numbers match the DestinationPrefix and SourcePrefix conditions. ▪ The manipulation rules are executed in the following order: RemoveFromLeft, RemoveFromRight, LeaveFromRight, Prefix2Add, and then Suffix2Add. ▪ Number Plan and Type can be used in the Remote-Party-ID header by configuring the EnableRPIHeader and AddTON2RPI parameters. ▪ To configure manipulation of destination numbers for Tel-to-IP calls using the Web interface, refer to "Configuring the Number Manipulation Tables" on page 143). ▪ For a description on using *ini* file table parameters, refer to "Configuring *ini* File Table Parameters" on page 222.
Web: Destination Phone Number Manipulation Table for IP to Tel Calls **EMS: EMS: SIP Manipulations > Destination IP to Telcom**	
[NumberMapIP2Tel]	This *ini* file table parameter manipulates the destination number of IP-to-Tel calls. The format of this parameter is as follows: [NumberMapIp2Tel] FORMAT NumberMapIp2Tel_Index = NumberMapIp2Tel_DestinationPrefix, NumberMapIp2Tel_SourcePrefix, NumberMapIp2Tel_SourceAddress, NumberMapIp2Tel_NumberType, NumberMapIp2Tel_NumberPlan, NumberMapIp2Tel_RemoveFromLeft, NumberMapIp2Tel_RemoveFromRight, NumberMapIp2Tel_LeaveFromRight, NumberMapIp2Tel_Prefix2Add, NumberMapIp2Tel_Suffix2Add, NumberMapIp2Tel_IsPresentationRestricted; [\NumberMapIp2Tel]

Parameter	Description
	For example: [NumberMapIp2Tel] NumberMapIp2Tel 0 = 01,034,10.13.77.8,$$,0,$$,2,$$,667,$$; NumberMapIp2Tel 1 = 10,10,1.1.1.1,255,255,3,0,5,100,$$,255; [\NumberMapIp2Tel] **Notes:** ▪ This table parameter can include up to 100 indices. ▪ The parameter IsPresentationRestricted is not applicable. ▪ RemoveFromLeft, RemoveFromRight, Prefix2Add, Suffix2Add, LeaveFromRight, NumberType, and NumberPlan are applied if the called and calling numbers match the DestinationPrefix, SourcePrefix, and SourceAddress conditions. ▪ The manipulation rules are executed in the following order: RemoveFromLeft, RemoveFromRight, LeaveFromRight, Prefix2Add, and then Suffix2Add. ▪ The Source IP address can include the 'x' wildcard to represent single digits. For example: 10.8.8.xx represents all addresses between 10.8.8.10 and 10.8.8.99. ▪ The Source IP address can include the asterisk ('*') wildcard to represent any number between 0 and 255. For example, 10.8.8.* represents all the addresses between 10.8.8.0 and 10.8.8.255. ▪ To configure manipulation of destination numbers for IP-to-Tel calls using the Web interface, refer to "Configuring the Number Manipulation Tables" on page 143). ▪ For a description on using *ini* file table parameters, refer to "Configuring *ini* File Table Parameters" on page 222.
Web: Source Phone Number Manipulation Table for Tel to IP Calls **EMS: SIP Manipulations > Source Telcom to IP**	
[SourceNumberMapTel2IP]	This *ini* file table parameter manipulates the source phone number for Tel-to-IP calls. The format of this parameter is as follows: [SourceNumberMapTel2Ip] FORMAT SourceNumberMapTel2Ip_Index = SourceNumberMapTel2Ip_DestinationPrefix, SourceNumberMapTel2Ip_SourcePrefix, SourceNumberMapTel2Ip_SourceAddress, SourceNumberMapTel2Ip_NumberType, SourceNumberMapTel2Ip_NumberPlan, SourceNumberMapTel2Ip_RemoveFromLeft, SourceNumberMapTel2Ip_RemoveFromRight, SourceNumberMapTel2Ip_LeaveFromRight, SourceNumberMapTel2Ip_Prefix2Add, SourceNumberMapTel2Ip_Suffix2Add, SourceNumberMapTel2Ip_IsPresentationRestricted,

Parameter	Description
	NumberMapTel2Ip_SrcTrunkGroupID, NumberMapTel2Ip_SrcIPGroupID; [\SourceNumberMapTel2Ip] For example: [SourceNumberMapTel2Ip] SourceNumberMapTel2Ip 0 = 22,03,$$,0,0,$$,2,$$,667,$$,0,$$,$$; SourceNumberMapTel2Ip 0 = 10,10,*,255,255,3,0,5,100,$$,255,$$,$$; [\SourceNumberMapTel2Ip] **Notes:** ▪ This table parameter can include up to 120 indices. ▪ RemoveFromLeft, RemoveFromRight, Prefix2Add, Suffix2Add, LeaveFromRight, NumberType, NumberPlan, and IsPresentationRestricted are applied if the called and calling numbers match the DestinationPrefix and SourcePrefix conditions. ▪ The manipulation rules are executed in the following order: RemoveFromLeft, RemoveFromRight, LeaveFromRight, Prefix2Add, and then Suffix2Add. ▪ An asterisk ('*') represents all IP addresses. ▪ IsPresentationRestricted is set to 'Restricted' only if 'Asserted Identity Mode' is set to 'P-Asserted'. ▪ Number Plan and Type can optionally be used in the Remote Party ID header by configuring the EnableRPIHeader and AddTON2RPI parameters. ▪ To configure manipulation of source numbers for Tel-to-IP calls using the Web interface, refer to "Configuring the Number Manipulation Tables" on page 143). ▪ For a description on using *ini* file table parameters, refer to to "Configuring *ini* File Table Parameters" on page 222.
Web: Source Phone Number Manipulation Table for IP to Tel Calls **EMS: EMS: SIP Manipulations > Source IP to Telcom**	
[SourceNumberMapIP2Tel]	This *ini* file table parameter manipulates the source number for IP-to-Tel calls. The format of this parameter is as follows: [SourceNumberMapIp2Tel] FORMAT SourceNumberMapIp2Tel_Index = SourceNumberMapIp2Tel_DestinationPrefix, SourceNumberMapIp2Tel_SourcePrefix, SourceNumberMapIp2Tel_SourceAddress, SourceNumberMapIp2Tel_NumberType, SourceNumberMapIp2Tel_NumberPlan, SourceNumberMapIp2Tel_RemoveFromLeft, SourceNumberMapIp2Tel_RemoveFromRight, SourceNumberMapIp2Tel_LeaveFromRight, SourceNumberMapIp2Tel_Prefix2Add, SourceNumberMapIp2Tel_Suffix2Add, SourceNumberMapIp2Tel_IsPresentationRestricted; [\SourceNumberMapIp2Tel]

Parameter	Description
	For example: [SourceNumberMapIp2Tel] SourceNumberMapIp2Tel 0 = 22,03,$$,$$,$$,$$,2,667,$$,$$; SourceNumberMapIp2Tel 1 = 034,01,1.1.1.1,$$,0,2,$$,$$,972,$$,10; [\SourceNumberMapIp2Tel] **Notes:** ▪ RemoveFromLeft, RemoveFromRight, Prefix2Add, Suffix2Add, LeaveFromRight, NumberType, and NumberPlan are applied if the called and calling numbers match the DestinationPrefix, SourcePrefix, and SourceAddress conditions. ▪ The manipulation rules are executed in the following order: RemoveFromLeft, RemoveFromRight, LeaveFromRight, Prefix2Add, and then Suffix2Add. ▪ The Source IP address can include the 'x' wildcard to represent single digits. For example: 10.8.8.xx represents all addresses between 10.8.8.10 and 10.8.8.99. ▪ The Source IP address can include the asterisk ('*') wildcard to represent any number between 0 and 255. For example, 10.8.8.* represents all the addresses between 10.8.8.0 and 10.8.8.255. ▪ To configure manipulation of source numbers for IP-to-Tel calls using the Web interface, refer to "Configuring the Number Manipulation Tables" on page 143). ▪ For a description on using *ini* file table parameters, refer to "Configuring *ini* File Table Parameters" on page 222.
For the ETSI ISDN variant, the following Number Plan and Type combinations (Plan/Type) are supported in the Destination and Source Manipulation tables: ▪ 0,0 = Unknown, Unknown ▪ 9,0 = Private, Unknown ▪ 9,1 = Private, Level 2 Regional ▪ 9,2 = Private, Level 1 Regional ▪ 9,3 = Private, PISN Specific ▪ 9,4 = Private, Level 0 Regional (local) ▪ 1,0 = Public(ISDN/E.164), Unknown ▪ 1,1 = Public(ISDN/E.164), International ▪ 1,2 = Public(ISDN/E.164), National ▪ 1,3 = Public(ISDN/E.164), Network Specific ▪ 1,4 = Public(ISDN/E.164), Subscriber ▪ 1,6 = Public(ISDN/E.164), Abbreviated For the NI-2 and DMS-100 ISDN variants, the valid combinations of TON and NPI for calling and called numbers are (Plan/Type): ▪ 0/0 - Unknown/Unknown ▪ 1/1 - International number in ISDN/Telephony numbering plan	

Parameter	Description
▪ 1/2 - National number in ISDN/Telephony numbering plan ▪ 1/4 - Subscriber (local) number in ISDN/Telephony numbering plan ▪ 9/4 - Subscriber (local) number in Private numbering plan	

6.18.4 LDAP Parameters

The Lightweight Directory Access Protocol (LDAP) parameters are described in the table below. For a detailed description on routing based on LDAP, refer to "Routing Based on LDAP Active Directory Queries" on page 537.

Table 6-64: LDAP Parameters

Parameter	Description
Web: LDAP Service **[LDAPServiceEnable]**	Determines whether to enable the LDAP service. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Note:** For this parameter to take effect, a device reset is required.
Web: LDAP Server IP **[LDAPServerIP]**	Defines the LDAP server's IP address in dotted-decimal notation (e.g., 192.10.1.255). The default is 0.0.0.0.
Web: LDAP Server Port **[LDAPServerPort]**	Defines the LDAP server's port number. The valid value range is 0 to 65535. The default port number is 389.
Web: LDAP Server Domain Name **[LDAPServerDomainName]**	Defines the host name of the LDAP server.
Web: LDAP Password **[LDAPPassword]**	Defines the LDAP server's user password.
Web: LDAP Bind DN **[LDAPBindDN]**	Defines the LDAP server's bind DN. This is used as the username during connection and binding to the server. For example: LDAPBindDN = "CN=Search user,OU=Labs,DC=OCSR2,DC=local"
Web: LDAP Search Dn **[LDAPSearchDN]**	Defines the search DN for LDAP search requests. This is the top DN of the subtree where the search is performed. This parameter is mandatory for the search. For example: LDAPSearchHDN = "CN=Search user,OU=Labs,DC=OCSR2,DC=local"
Web: LDAP Server Max Respond Time **[LDAPServerMaxRespondTime]**	Defines the time (in seconds) that the device waits for LDAP server responses. The valid value range is 0 to 86400. The default is 3000.
[LDAPDebugMode]	Determines whether to enable the LDAP task debug messages. This is used for providing debug information regarding LDAP tasks. The valid value range is 0 to 3. The default is 0.

Parameter	Description
Web: MS LDAP OCS Number attribute name **[MSLDAPOCSNumAttributeName]**	The name of the attribute that represents the user OCS number in the Microsoft AD database. The valid value is a string of up to 49 characters. The default is "msRTCSIP-PrimaryUserAddress".
Web: MS LDAP PBX Number attribute name **[MSLDAPPBXNumAttributeName]**	The name of the attribute that represents the user PBX number in the Microsoft AD database. The valid value is a string of up to 49 characters. The default is "telephoneNumber".
Web: MS LDAP MOBILE Number attribute name **[MSLDAPMobileNumAttributeName]**	The name of the attribute that represents the user Mobile number in the Microsoft AD database. The valid value is a string of up to 49 characters. The default is "mobile".

6.19 Channel Parameters

This subsection describes the device's channel parameters.

6.19.1 Voice Parameters

The voice parameters are described in the table below.

Table 6-65: Voice Parameters

Parameter	Description
Web/EMS: Input Gain **[InputGain]**	Pulse-code modulation (PCM) input gain control (in decibels). This parameter sets the level for the received (Tel/PSTN-to-IP) signal. The valid range is -32 to 31 dB. The default value is 0 dB.
Web: Voice Volume EMS: Volume (dB) **[VoiceVolume]**	Voice gain control (in decibels). This parameter sets the level for the transmitted (IP-to-Tel/PSTN) signal. The valid range is -32 to 31 dB. The default value is 0 dB.
EMS: Payload Format **[VoicePayloadFormat]**	Determines the bit ordering of the G.726/G.727 voice payload format. ▪ **[0]** = Little Endian (default) ▪ **[1]** = Big Endian **Note:** To ensure high voice quality when using G.726/G.727, both communicating ends should use the same endianness format. Therefore, when the device communicates with a third-party entity that uses the G.726/G.727 voice coder and voice quality is poor, change the settings of this parameter (between Big Endian and Little Endian).
Web: MF Transport Type **[MFTransportType]**	Currently, not supported.
Web: Enable Answer Detector **[EnableAnswerDetector]**	Currently, not supported.

Parameter	Description
Web: Answer Detector Activity Delay **[AnswerDetectorActivityDelay]**	Determines (in 100-msec resolution) the time between sending Activation command (AnswerDetector = 1) and the time that the detector actually starts to work. The valid range is 0 to 1023. The default is 0.
Web: Answer Detector Silence Time **[AnswerDetectorSilenceTime]**	Currently, not supported.
Web: Answer Detector Redirection **[AnswerDetectorRedirection]**	Currently, not supported.
Web: Answer Detector Sensitivity EMS: Sensitivity **[AnswerDetectorSensitivity]**	Determines the Answer Detector sensitivity. The range is 0 (most sensitive) to 2 (least sensitive). The default is 0.
Web: Silence Suppression EMS: Silence Compression Mode **[EnableSilenceCompression]**	Silence Suppression is a method for conserving bandwidth on VoIP calls by not sending packets when silence is detected. ▪ **[0]** Disable = Silence Suppression is disabled (default). ▪ **[1]** Enable = Silence Suppression is enabled. ▪ **[2]** Enable without Adaptation = A single silence packet is sent during a silence period (applicable only to G.729). **Note:** If the selected coder is G.729, the value of the 'annexb' parameter of the fmtp attribute in the SDP is determined by the following rules: ▪ If EnableSilenceCompression is 0: 'annexb=no'. ▪ If EnableSilenceCompression is 1: 'annexb=yes'. ▪ If EnableSilenceCompression is 2 and IsCiscoSCEMode is 0: 'annexb=yes'. ▪ If EnableSilenceCompression is 2 and IsCiscoSCEMode is 1: 'annexb=no'.
Web: Echo Canceler EMS: Echo Canceller Enable **[EnableEchoCanceller]**	Determines whether echo cancellation is enabled and therefore, echo from voice calls is removed. ▪ **[0]** Off = Echo Canceler is disabled. ▪ **[1]** On = Echo Canceler is enabled (default). **Note:** This parameter is used to maintain backward compatibility.
Web: Max Echo Canceller Length **[MaxEchoCancellerLength]**	Determines the maximum Echo Canceler Length (in msec), which is the maximum echo path delay (tail length) for which the echo canceller is designed to operate: ▪ **[0]** Default = based on various internal device settings to attain maximum channel capacity (default) ▪ **[11]** 64 msec ▪ **[22]** 128 msec **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ Using 128 msec reduces the channel capacity to 200 channels. ▪ When housed with an analog/BRI module, the device (Mediant 1000) can use a max. echo canceller length of 64 msec.

Parameter	Description
	▪ When housed with PRI TRUNKS module, the device (Mediant 1000) can use a max. echo canceller length of 128 msec. ▪ When set to 128 msec, the number of available channels is reduced by a factor of 5/6 (for Mediant 1000). ▪ When housed with an MPM module (in Slot #6), no channel reduction occurs (for Mediant 1000). ▪ It is unnecessary to configure the parameter EchoCancellerLength, as it automatically acquires its value from this parameter.
EMS: Echo Canceller Hybrid Loss **[ECHybridLoss]**	Sets the four wire to two wire worst-case Hybrid loss, the ratio between the signal level sent to the hybrid and the echo level returning from the hybrid. ▪ **[0]** = 6 dB (default) ▪ **[1]** = N/A ▪ **[2]** = 0 dB ▪ **[3]** = 3 dB
[ECNLPMode]	Defines the echo cancellation Non-Linear Processing (NLP) mode. ▪ **[0]** = NLP adapts according to echo changes (default). ▪ **[1]** = Disables NLP.
[EchoCancellerAggressiveNLP]	Enables or disables the Aggressive NLP at the first 0.5 second of the call. When enabled, the echo is removed only in the first half of a second of the incoming IP signal. ▪ **[0]** = Disable ▪ **[1]** = Enable (default) **Note:** For this parameter to take effect, a device reset is required.
[EnableNoiseReduction]	Enables or disables the DSP Noise Reduction mechanism. ▪ **[0]** = Disable (default). ▪ **[1]** = Enable. **Note:** When this parameter is enabled the channel capacity might be reduced.
Web: Enable RFC 3389 CN Payload Type EMS: Comfort Noise Enable **[EnableStandardSIDPayloadType]**	Determines whether Silence Indicator (SID) packets are sent according to RFC 3389. ▪ **[0]** Disable = G.711 SID packets are sent in a proprietary method (default). ▪ **[1]** Enable = SID (comfort noise) packets are sent with the RTP SID payload type according to RFC 3389. This is applicable only to G.711 and G.726 coders.
[RTPSIDCoeffNum]	Determines the number of spectral coefficients added to an SID packet being sent according to RFC 3389. Valid only if EnableStandardSIDPayloadType is set to 1. The valid values are **[0]** (default), **[4]**, **[6]**, **[8]** and **[10]**.

6.19.2 Coder Parameters

The coder parameters are described in the table below.

Table 6-66: Coder Parameters

Parameter	Description
Web: Enable RFC 4117 Transcoding **[EnableRFC4117Transcoding]**	Enables transcoding of calls according to RFC 4117. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ For a detailed description of this transcoding feature, refer to Transcoding using Third-Party Call Control on page 527.
[EnableEVRCVAD]	Enables or disables the EVRC voice activity detector. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable **Note:** Supported for EVRC and EVRC-B coders.
EMS: VBR Coder DTX Min **[EVRCDTXMin]**	Defines the minimum gap between two SID frames when using the EVRC voice activity detector. Units are in EVRC frame size (20 msec). The range is 0 to 20000. The default value is 12. **Note:** Supported for EVRC and EVRC-B coders.
EMS: VBR Coder DTX Max **[EVRCDTXMax]**	Defines the maximum gap between two SID frames when using the EVRC voice activity detector. Units are in EVRC frame size (20 msec). The range is 0 to 20000. The default value is 32. **Note:** This parameter is applicable only to EVRC and EVRC-B coders.
Web: DSP Version Template Number EMS: Version Template Number **[DSPVersionTemplateNumber]**	Determines the DSP template to use on the device. Each DSP template supports specific coders, channel capacity, and features. For the list of supported DSP templates, refer to the device's Release Notes. The default is DSP template 0. You can load different DSP templates to analog and digital modules, using the syntax DSPVersionTemplateNumber=xy where: ▪ x = 0 or 1 for DSP templates of analog modules ▪ y = 0 to 5 for DSP templates of digital and MPM modules **Note:** For this parameter to take effect, a device reset is required.
Web: DSP Template Mix Table **EMS: VoP Media Provisioning > General** Settings	
[DspTemplates]	This *ini* file table parameter allows the device to use a combination of up to two DSP templates and determines the percentage of DSP resources allocated per DSP template. The DSP templates' values and capabilities (i.e., supported coders, channel capacity, and features) are specified in the device's Release Notes. The format of this table is as follows:

Parameter	Description
	[DspTemplates] FORMAT DspTemplates_Index = DspTemplates_DspTemplateNumber, DspTemplates_DspResourcesPercentage; [\DspTemplates] For example, to load DSP Template 1 to 50% of the DSPs, and DSP Template 2 to the remaining 50%, the table is configured as follows: DspTemplates 0 = 1, 50; DspTemplates 1 = 2, 50; **Note:** The *ini* file parameter DSPVersionTemplateNumber is ignored when using the parameters specified in this table.
EMS: VBR Coder Header Format **[VBRCoderHeaderFormat]**	Defines the format of the RTP header for VBR coders. ▪ **[0]** = Payload only (no header, TOC, or m-factor) - similar to RFC 3558 Header Free format (default). ▪ **[1]** = Supports RFC 2658 - 1 byte for interleaving header (always 0), TOC, no m-factor. ▪ **[2]** = Payload including TOC only, allow m-factor. ▪ **[3]** = RFC 3558 Interleave/Bundled format.
EMS: VBR Coder Hangover **[VBRCoderHangover]**	Determines the required number of silence frames at the beginning of each silence period when using the VBR coder silence suppression. The range is 0 to 255. The default value is 1.

6.19.3 Fax and Modem Parameters

The fax and modem parameters are described in the table below.

Table 6-67: Fax and Modem Parameters

Parameter	Description
Web: Fax Transport Mode EMS: Transport Mode **[FaxTransportMode]**	Fax transport mode used by the device. ▪ **[0]** Disable = transparent mode. ▪ **[1]** T.38 Relay = (default). ▪ **[2]** Bypass. ▪ **[3]** Events Only. **Note:** This parameter is overridden by the parameter IsFaxUsed. If the parameter IsFaxUsed is set to 1 (T.38 Relay) or 3 (Fax Fallback), then FaxTransportMode is always set to 1 (T.38 relay).
Web: Fax Relay Enhanced Redundancy Depth EMS: Enhanced Relay Redundancy Depth **[FaxRelayEnhancedRedundancyDepth]**	Number of times that control packets are retransmitted when using the T.38 standard. The valid range is 0 to 4. The default value is 2.

Parameter	Description
Web: Fax Relay Redundancy Depth EMS: Relay Redundancy Depth **[FaxRelayRedundancyDepth]**	Number of times that each fax relay payload is retransmitted to the network. ▪ **[0]** = No redundancy (default). ▪ **[1]** = One packet redundancy. ▪ **[2]** = Two packet redundancy. **Note:** This parameter is applicable only to non-V.21 packets.
Web: Fax Relay Max Rate (bps) EMS: Relay Max Rate **[FaxRelayMaxRate]**	Maximum rate (in bps) at which fax relay messages are transmitted (outgoing calls). ▪ **[0]** 2400 = 2.4 kbps ▪ **[1]** 4800 = 4.8 kbps ▪ **[2]** 7200 = 7.2 kbps ▪ **[3]** 9600 = 9.6 kbps ▪ **[4]** 12000 = 12.0 kbps ▪ **[5]** 14400 = 14.4 kbps (default) ▪ **[6]** 16800bps = 16.8 kbps ▪ **[7]** 19200bps = 19.2 kbps ▪ **[8]** 21600bps = 21.6 kbps ▪ **[9]** 24000bps = 24 kbps ▪ **[10]** 26400bps = 26.4 kbps ▪ **[11]** 28800bps = 28.8 kbps ▪ **[12]** 31200bps = 31.2 kbps ▪ **[13]** 33600bps = 33.6 kbps **Notes:** ▪ The rate is negotiated between both sides (i.e., the device adapts to the capabilities of the remote side). ▪ Configuration above 14.4 kbps is truncated to 14.4 kbps for non-T.38 V.34 supporting <devices>.
Web: Fax Relay ECM Enable EMS: Relay ECM Enable **[FaxRelayECMEnable]**	Determines whether the Error Correction Mode (ECM) mode is used during fax relay. ▪ **[0]** Disable = ECM mode is not used during fax relay. ▪ **[1]** Enable = ECM mode is used during fax relay (default).
Web: Fax/Modem Bypass Coder Type EMS: Coder Type **[FaxModemBypassCoderType]**	Coder used by the device when performing fax/modem bypass. Usually, high-bit-rate coders such as G.711 should be used. ▪ **[0]** G.711Alaw= G.711 A-law 64 (default). ▪ **[1]** G.711Mulaw = G.711 μ-law.

Parameter	Description
Web/EMS: CNG Detector Mode **[CNGDetectorMode]**	Determines whether the device detects the fax Calling tone (CNG). ▪ **[0]** Disable = The originating device doesn't detect CNG; the CNG signal passes transparently to the remote side (default). ▪ **[1]** Relay = CNG is detected on the originating side. CNG packets are sent to the remote side according to T.38 (if IsFaxUsed = 1) and the fax session is started. A SIP Re-INVITE message isn't sent and the fax session starts by the terminating device. This option is useful, for example, when the originating device is located behind a firewall that blocks incoming T.38 packets on ports that have not yet received T.38 packets from the internal network (i.e., originating device). To also send a Re-INVITE message upon detection of a fax CNG tone in this mode, set the parameter FaxCNGMode to 1. ▪ **[2]** Events Only = CNG is detected on the originating side and a fax session is started by the originating side using the Re-INVITE message. Usually, T.38 fax session starts when the 'preamble' signal is detected by the answering side. Some SIP devices don't support the detection of this fax signal on the answering side and thus, in these cases it is possible to configure the device to start the T.38 fax session when the CNG tone is detected by the originating side. However, this mode is not recommended.
Web: Fax/Modem Bypass Packing Factor EMS: Packetization Period **[FaxModemBypassM]**	Number of (20 msec) coder payloads that are used to generate a fax/modem bypass packet. The valid range is 1, 2, or 3 coder payloads. The default value is 1 coder payload.
[FaxModemNTEMode]	Determines whether the device sends RFC 2833 ANS/ANSam events upon detection of fax and/or modem Answer tones (i.e., CED tone). ▪ **[0]** = Disabled (default). ▪ **[1]** = Enabled. **Note:** This parameter is applicable only when the fax or modem transport type is set to bypass or Transparent-with-Events.
Web/EMS: Fax Bypass Payload Type **[FaxBypassPayloadType]**	Determines the fax bypass RTP dynamic payload type. The valid range is 96 to 120. The default value is 102.
EMS: Modem Bypass Payload Type **[ModemBypassPayloadType]**	Modem Bypass dynamic payload type. The range is 0-127. The default value is 103.

Parameter	Description
EMS: Relay Volume (dBm) **[FaxModemRelayVolume]**	Determines the fax gain control. The range is -18 to -3, corresponding to -18 dBm to -3 dBm in 1-dB steps. The default is -6 dBm fax gain control.
Web/EMS: Fax Bypass Output Gain **[FaxBypassOutputGain]**	Defines the fax bypass output gain control. The range is -31 to +31 dB, in 1-dB steps. The default is 0 (i.e., no gain).
Web/EMS: Modem Bypass Output Gain **[ModemBypassOutputGain]**	Defines the modem bypass output gain control. The range is -31 dB to +31 dB, in 1-dB steps. The default is 0 (i.e., no gain).
EMS: NTE Max Duration **[NTEMaxDuration]**	Maximum time for sending Named Telephony Events (NTEs) to the IP side regardless of the time range when the TDM signal is detected. The range is -1 to 200,000,000 msec (i.e., 55 hours). The default is -1 (i.e., NTE stops only upon detection of an End event).
EMS: Basic Packet Interval **[FaxModemBypassBasicRTPPacketInterval]**	Determines the basic frame size that is used during fax/modem bypass sessions. ▪ **[0]** = Determined internally (default) ▪ **[1]** = 5 msec (not recommended) ▪ **[2]** = 10 msec ▪ **[3]** = 20 msec **Note:** When set to 5 msec (1), the maximum number of simultaneous channels supported is 120.
EMS: Dynamic Jitter Buffer Minimal Delay (dB) **[FaxModemBypassDJBufMinDelay]**	Determines the Jitter Buffer delay (in milliseconds) during fax and modem bypass session. The range is 0 to 150 msec. The default is 40.
EMS: Enable Inband Network Detection **[EnableFaxModemInbandNetworkDetection]**	Enables or disables in-band network detection related to fax/modem. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable When this parameter is enabled on Bypass and transparent with events mode (VxxTransportType = 2 or 3), a detection of an Answer Tone from the network triggers a switch to bypass mode in addition to the local Fax/Modem tone detections. However, only a high bit-rate coder voice session effectively detects the Answer Tone sent by a remote endpoint. This can be useful when, for example, the payload of voice and bypass is the same, allowing the originator to switch to bypass mode as well.

Parameter	Description
EMS: NSE Mode **[NSEMode]**	Cisco compatible fax and modem bypass mode. ▪ **[0]** = NSE disabled (default) ▪ **[1]** = NSE enabled **Notes:** ▪ This feature can be used only if VxxModemTransportType = 2 (Bypass). ▪ If NSE mode is enabled, the SDP contains the following line: 'a=rtpmap:100 X-NSE/8000'. ▪ To use this feature: ✓ The Cisco gateway must include the following definition: 'modem passthrough nse payload-type 100 codec g711alaw'. ✓ Set the Modem transport type to Bypass mode (VxxModemTransportType = 2) for all modems. ✓ Configure the gateway parameter NSEPayloadType = 100. In NSE bypass mode, the device starts using G.711 A-Law (default) or G.711μ-Law according to the parameter FaxModemBypassCoderType. The payload type used with these G.711 coders is a standard one (8 for G.711 A-Law and 0 for G.711 μ-Law). The parameters defining payload type for the 'old' Bypass mode FaxBypassPayloadType and ModemBypassPayloadType are not used with NSE Bypass. The bypass packet interval is selected according to the parameter FaxModemBypassBasicRtpPacketInterval.
EMS: NSE Payload Type **[NSEPayloadType]**	NSE payload type for Cisco Bypass compatible mode. The valid range is 96-127. The default value is 105. **Note:** Cisco gateways usually use NSE payload type of 100.
Web: V.21 Modem Transport Type EMS: V21 Transport **[V21ModemTransportType]**	V.21 Modem Transport Type used by the device. ▪ **[0]** Disable = Disable (Transparent) - default ▪ **[1]** Enable Relay = N/A ▪ **[2]** Enable Bypass. ▪ **[3]** Events Only = Transparent with Events
Web: V.22 Modem Transport Type EMS: V22 Transport **[V22ModemTransportType]**	V.22 Modem Transport Type used by the device. ▪ **[0]** Disable = Disable (Transparent) ▪ **[1]** Enable Relay = N/A ▪ **[2]** Enable Bypass = (default) ▪ **[3]** Events Only = Transparent with Events
Web: V.23 Modem Transport Type EMS: V23 Transport **[V23ModemTransportType]**	V.23 Modem Transport Type used by the device. ▪ **[0]** Disable = Disable (Transparent)

Parameter	Description
	▪ **[1]** Enable Relay = N/A ▪ **[2]** Enable Bypass = (default) ▪ **[3]** Events Only = Transparent with Events
Web: V.32 Modem Transport Type EMS: V32 Transport **[V32ModemTransportType]**	V.32 Modem Transport Type used by the device. ▪ **[0]** Disable = Disable (Transparent) ▪ **[1]** Enable Relay = N/A ▪ **[2]** Enable Bypass = (default) ▪ **[3]** Events Only = Transparent with Events **Note:** This parameter applies only to V.32 and V.32bis modems.
Web: V.34 Modem Transport Type EMS: V34 Transport **[V34ModemTransportType]**	V.90/V.34 Modem Transport Type used by the device. ▪ **[0]** Disable = Disable (Transparent) ▪ **[1]** Enable Relay = N/A ▪ **[2]** Enable Bypass = (default) ▪ **[3]** Events Only = Transparent with Events
EMS: Bell Transport Type **[BellModemTransportType]**	Determines the Bell modem transport method. ▪ **[0]** = Transparent (default). ▪ **[2]** = Bypass. ▪ **[3]** = Transparent with events.

6.19.4 DTMF Parameters

The dual-tone multi-frequency (DTMF) parameters are described in the table below.

Table 6-68: DTMF Parameters

Parameter	Description
Web/EMS: DTMF Transport Type **[DTMFTransportType]**	Determines the DTMF transport type. ▪ **[0]** DTMF Mute = Erases digits from voice stream and doesn't relay to remote. ▪ **[2]** Transparent DTMF = Digits remain in voice stream. ▪ **[3]** RFC 2833 Relay DTMF = Erases digits from voice stream and relays to remote according to RFC 2833 (default). ▪ **[7]** RFC 2833 Relay Rcv Mute = DTMFs are sent according to RFC 2833 and muted when received. **Note:** This parameter is automatically updated if the parameters TxDTMFOption or RxDTMFOption are configured.
Web: DTMF Volume (-31 to 0 dB) EMS: DTMF Volume (dBm) **[DTMFVolume]**	DTMF gain control value (in decibels) to the PSTN or analog side. The valid range is -31 to 0 dB. The default value is -11 dB.
Web: DTMF Generation Twist EMS: DTMF Twist Control **[DTMFGenerationTwist]**	Defines the range (in decibels) between the high and low frequency components in the DTMF signal. Positive decibel values cause the higher frequency component to be stronger than the lower one. Negative values cause the opposite effect. For any parameter value, both components change so that their average is constant. The valid range is -10 to 10 dB. The default value is 0 dB. **Note:** For this parameter to take effect, a device reset is required.
EMS: DTMF Inter Interval (msec) **[DTMFInterDigitInterval]**	Time in msec between generated DTMF digits to PSTN side (if TxDTMFOption = 1, 2 or 3). The default value is 100 msec. The valid range is 0 to 32767.
EMS: DTMF Length (msec) **[DTMFDigitLength]**	Time (in msec) for generating DTMF tones to the PSTN side (if TxDTMFOption = 1, 2 or 3). It also configures the duration that is sent in INFO (Cisco) messages. The valid range is 0 to 32767. The default value is 100.
EMS: Rx DTMF Relay Hang Over Time (msec) **[RxDTMFHangOverTime]**	Defines the Voice Silence time (in msec) after playing DTMF or MF digits to the Tel/PSTN side that arrive as Relay from the IP side. Valid range is 0 to 2,000 msec. The default is 1,000 msec.
EMS: Tx DTMF Relay Hang Over Time (msec) **[TxDTMFHangOverTime]**	Defines the Voice Silence time (in msec) after detecting the end of DTMF or MF digits at the Tel/PSTN side when the DTMF Transport Type is either Relay or Mute. Valid range is 0 to 2,000 msec. The default is 1,000 msec.

Parameter	Description
Web: Enable Special Digits EMS: Use '#' For Dial Termination **[IsSpecialDigits]**	Determines whether the asterisk (*) and pound (#) digits can be used in DTMF. ▪ **[0]** Disable = Use '*' or '#' to terminate number collection (refer to the parameter UseDigitForSpecialDTMF). (Default.) ▪ **[1]** Enable = Allows '*' and '#' for telephone numbers dialed by a user or for the endpoint telephone number. **Note:** These symbols can always be used as the first digit of a dialed number even if you disable this parameter.

6.19.5 RTP, RTCP and T.38 Parameters

The RTP, RTCP and T.38 parameters are described in the table below.

Table 6-69: RTP/RTCP and T.38 Parameters

Parameter	Description
Web: Dynamic Jitter Buffer Minimum Delay EMS: Minimal Delay (dB) **[DJBufMinDelay]**	Minimum delay (in msec) for the Dynamic Jitter Buffer. The valid range is 0 to 150. The default delay is 10. **Note:** For more information on Jitter Buffer, refer to "Dynamic Jitter Buffer Operation" on page 602.
Web: Dynamic Jitter Buffer Optimization Factor EMS: Opt Factor **[DJBufOptFactor]**	Dynamic Jitter Buffer frame error/delay optimization factor. The valid range is 0 to 13. The default factor is 10. **Notes:** ▪ For data (fax and modem) calls, set this parameter to 13. ▪ For more information on Jitter Buffer, refer to "Dynamic Jitter Buffer Operation" on page 602.
Web/EMS: Analog Signal Transport Type **[AnalogSignalTransportType]**	Determines the analog signal transport type. ▪ **[0]** Ignore Analog Signals = Ignore (default). ▪ **[1]** RFC 2833 Analog Signal Relay = Transfer hookflash using RFC 2833.
Web: RTP Redundancy Depth EMS: Redundancy Depth **[RTPRedundancyDepth]**	Determines whether the device generates redundant packets. This can be used for packet loss where the missing information (audio) can be reconstructed at the receiver end from the redundant data that arrives in the subsequent packet(s). ▪ **[0]** 0 = Disable the generation of redundant packets (default). ▪ **[1]** 1 = Enable the generation of RFC 2198 redundancy packets (payload type defined by the parameter RFC2198PayloadType). **Note:** The RTP redundancy dynamic payload type can be included in the SDP, by using the parameter EnableRTPRedundancyNegotiation.

Parameter	Description
Web: Enable RTP Redundancy Negotiation **[EnableRTPRedundancyNegotiation]**	Determines whether the device includes the RTP redundancy dynamic payload type in the SDP, according to RFC 2198. ▪ **[0]** Disable (default) ▪ **[1]** Enable When enabled, the device includes in the SDP message the RTP payload type "RED" and the payload type configured by the parameter RFC2198PayloadType. `a=rtpmap:<PT> RED/8000` Where *<PT>* is the payload type as defined by RFC2198PayloadType. The device sends the INVITE message with "a=rtpmap:<PT> RED/8000" and responds with a 18x/200 OK and "a=rtpmap:<PT> RED/8000" in the SDP. **Notes:** ▪ For this feature to be functional, you must also set the parameter RTPRedundancyDepth to 1 (i.e., enabled). ▪ Currently, the negotiation of “RED” payload type is not supported and therefore, it should be configured to the same PT value for both parties.
Web: RFC 2198 Payload Type EMS: Redundancy Payload Type **[RFC2198PayloadType]**	RTP redundancy packet payload type according to RFC 2198. The range is 96 to 127. The default is 104. **Note:** This parameter is applicable only if the parameter RTPRedundancyDepth is set to 1.
Web: Packing Factor EMS: Packetization Factor **[RTPPackingFactor]**	N/A. Controlled internally by the device according to the selected coder.
Web/EMS: Basic RTP Packet Interval **[BasicRTPPacketInterval]**	N/A. Controlled internally by the device according to the selected coder.
Web: RTP Directional Control **[RTPDirectionControl]**	N/A. Controlled internally by the device according to the selected coder.
Web/EMS: RFC 2833 TX Payload Type **[RFC2833TxPayloadType]**	N/A. Use the *ini* file parameter RFC2833PayloadType instead.
Web/EMS: RFC 2833 RX Payload Type **[RFC2833RxPayloadType]**	N/A. Use the *ini* file parameter RFC2833PayloadType instead.
[EnableDetectRemoteMACChange]	Changes the RTP packets according to the MAC address of received RTP packets and according to Gratuitous Address Resolution Protocol (GARP) messages. ▪ **[0]** = Nothing is changed. ▪ **[1]** = If the device receives RTP packets with a different source MAC address (than the MAC address of the transmitted RTP packets), then it sends RTP packets to this MAC address and removes this IP entry from the device's ARP cache table.

Parameter	Description
	▪ **[2]** = The device uses the received GARP packets to change the MAC address of the transmitted RTP packets (default). ▪ **[3]** = Options 1 and 2 are used. **Note:** For this parameter to take effect, a device reset is required.
Web: RTP Base UDP Port EMS: Base UDP Port **[BaseUDPport]**	Lower boundary of the UDP port used for RTP, RTCP (RTP port + 1) and T.38 (RTP port + 2). The upper boundary of the UDP port range is the Base UDP Port + 10 * number of the device's channels. The range of possible UDP ports is 6,000 to 64,000. The default base UDP port is 6000. For example, if the Base UDP Port is set to 6000, then 1) one channel may use the ports RTP 6000, RTCP 6001, and T.38 6002, 2) another channel may use RTP 6010, RTCP 6011, and T.38 6012, etc. The UDP port range is as follows: ▪ BaseUDPport to BaseUDPport + 255*10 **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ The UDP ports are allocated randomly to channels. ▪ You can define a UDP port range per Media Realm (refer to "Configuring Media Realms" on page 99). ▪ If RTP Base UDP Port is not a factor of 10, the following message is generated: 'invalid local RTP port'. ▪ For detailed information on the default RTP/RTCP/T.38 port allocation, refer to the *Product Reference Manual*.
Web: Remote RTP Base UDP Port EMS: Remote Base UDP Port **[RemoteBaseUDPPort]**	Determines the lower boundary of UDP ports used for RTP, RTCP and T.38 by a remote device. If this parameter is set to a non-zero value, ThroughPacket™ (RTP multiplexing) is enabled. The device uses this parameter (and BaseUDPPort) to identify and distribute the payloads from the received multiplexed IP packet to the relevant channels. The valid range is the range of possible UDP ports: 6,000 to 64,000. The default value is 0 (i.e., RTP multiplexing is disabled). For detailed information on RTP multiplexing, refer to RTP Multiplexing (ThroughPacket) on page 601. **Notes:** ▪ The value of this parameter on the local device must equal the value of BaseUDPPort on the remote device. ▪ To enable RTP multiplexing, the parameters L1L1ComplexTxUDPPort and L1L1ComplexRxUDPPort must be set to a non-zero value. ▪ When VLANs are implemented, RTP multiplexing is not supported.

Parameter	Description
Web: RTP Multiplexing Local UDP Port **[L1L1ComplexTxUDPPort]**	Determines the local UDP port used for outgoing multiplexed RTP packets (applies to RTP multiplexing). The valid range is the range of possible UDP ports: 6,000 to 64,000. The default value is 0 (i.e., RTP multiplexing is disabled). **Note:** For this parameter to take effect, a device reset is required.
Web: RTP Multiplexing Remote UDP Port **[L1L1ComplexRxUDPPort]**	Determines the remote UDP port to where the multiplexed RTP packets are sent and the local UDP port used for incoming multiplexed RTP packets (applies to RTP multiplexing). The valid range is the range of possible UDP ports: 6,000 to 64,000. The default value is 0 (i.e., RTP multiplexing is disabled). **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ All devices that participate in the same RTP multiplexing session must use this same port.
EMS: No Op Enable **[NoOpEnable]**	Enables or disables the transmission of RTP or T.38 No-Op packets. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable This mechanism ensures that the NAT binding remains open during RTP or T.38 silence periods.
EMS: No Op Interval **[NoOpInterval]**	Defines the time interval in which RTP or T.38 No-Op packets are sent in the case of silence (no RTP/T.38 traffic) when No-Op packet transmission is enabled. The valid range is 20 to 65,000 msec. The default is 10,000. **Note:** To enable No-Op packet transmission, use the NoOpEnable parameter.
EMS: No Op Payload Type **[RTPNoOpPayloadType]**	Determines the payload type of No-Op packets. The valid range is 96 to 127 (for the range of Dynamic RTP Payload Type for all types of non hard-coded RTP Payload types, refer to RFC 3551). The default value is 120. **Note:** When defining this parameter, ensure that it doesn't cause collision with other payload types.
RTCP XR Settings (**Note:** For a detailed description of RTCP XR reports, refer to the Product Reference Manual.)	
Web: Enable RTCP XR EMS: RTCP XR Enable **[VQMonEnable]**	Enables voice quality monitoring and RTCP Extended Reports (RTCP XR). ▪ **[0]** Disable = Disable (default) ▪ **[1]** Enable = Enables **Note:** For this parameter to take effect, a device reset is required.

Parameter	Description
Web: Minimum Gap Size EMS: GMin **[VQMonGMin]**	Voice quality monitoring - minimum gap size (number of frames). The default is 16.
Web/EMS: Burst Threshold **[VQMonBurstHR]**	Voice quality monitoring - excessive burst alert threshold. if set to -1 (default), no alerts are issued.
Web/EMS: Delay Threshold **[VQMonDelayTHR]**	Voice quality monitoring - excessive delay alert threshold. if set to -1 (default), no alerts are issued.
Web: R-Value Delay Threshold EMS: End of Call Rval Delay Threshold **[VQMonEOCRValTHR]**	Voice quality monitoring - end of call low quality alert threshold. if set to -1 (default), no alerts are issued.
Web: RTCP Packet Interval EMS: Packet Interval **[RTCPInterval]**	Defines the time interval (in msec) between adjacent RTCP reports. The interval range is 0 to 65,535. The default interval is 5,000.
Web: Disable RTCP Interval Randomization EMS: Disable Interval Randomization **[DisableRTCPRandomize]**	Controls whether RTCP report intervals are randomized or whether each report interval accords exactly to the parameter RTCPInterval. ▪ **[0]** Disable = Randomize (default) ▪ **[1]** Enable = No Randomize
EMS: Esc Transport Type **[RTCPXRESCTransportType]**	Determines the transport layer used for outgoing SIP dialogs initiated by the device to the RTCP-XR Collection Server. ▪ **[-1]** Not Configured (default) ▪ **[0]** UDP ▪ **[1]** TCP ▪ **[2]** TLS **Note:** When set to 'Not Configured', the value of the parameter SIPTransportType is used.
Web: RTCP XR Collection Server EMS: Esc IP **[RTCPXREscIP]**	IP address of the Event State Compositor (ESC). The device sends RTCP XR reports to this server, using PUBLISH messages. The address can be configured as a numerical IP address or as a domain name.
Web: RTCP XR Report Mode EMS: Report Mode **[RTCPXRReportMode]**	Determines whether RTCP XR reports are sent to the Event State Compositor (ESC), and if so, defines the interval in which they are sent. ▪ **[0]** Disable = RTCP XR reports are not sent to the ESC (default). ▪ **[1]** End Call = RTCP XR reports are sent to the ESC at the end of each call. ▪ **[2]** End Call & Periodic = RTCP XR reports are sent to the ESC at the end of each call and periodically according to the parameter RTCPInterval.

6.20 Auxiliary and Configuration Files Parameters

This subsection describes the device's auxiliary and configuration files parameters.

6.20.1 Auxiliary/Configuration File Name Parameters

The configuration files (i.e., auxiliary files) can be loaded to the device using the Web interface or a TFTP session (refer to "Loading Auxiliary Files" on page 200). For loading these files using the *ini* file, you need to configure these files in the *ini* file and configured whether they must be stored in the non-volatile memory. The table below lists the *ini* file parameters associated with these auxiliary files. For a detailed description of the auxiliary files, refer to "Auxiliary Configuration Files" on page 473.

Table 6-70: Auxiliary and Configuration File Parameters

Parameter	Description
General Parameters	
[SetDefaultOnIniFileProcess]	Determines if all the device's parameters are set to their defaults before processing the updated *ini* file. ▪ **[0]** Disable - parameters not included in the downloaded *ini* file are not returned to default settings (i.e., retain their current settings). ▪ **[1]** Enable (default) **Note:** This parameter is applicable only for automatic HTTP update or Web *ini* file upload (not applicable if the *ini* file is loaded using BootP).
[SaveConfiguration]	Determines if the device's configuration (parameters and files) is saved to flash (non-volatile memory). ▪ **[0]** = Configuration isn't saved to flash memory. ▪ **[1]** = Configuration is saved to flash memory (default).
Auxiliary and Configuration File Name Parameters	
Web/EMS: Call Progress Tones File **[CallProgressTonesFilename]**	The name of the file containing the Call Progress Tones definitions. Refer to the *Product Reference Manual* for additional information on how to create and load this file. **Note:** For this parameter to take effect, a device reset is required.
Web/EMS: Voice Prompts File **[VoicePromptsFileName]**	The name (and path) of the file containing the Voice Prompts. For a detailed description of this file, refer to Voice Prompts File on page 479. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ This parameter is applicable only to Mediant 1000.
Web/EMS: Prerecorded Tones File **[PrerecordedTonesFileName]**	The name (and path) of the file containing the Prerecorded Tones. **Note:** For this parameter to take effect, a device reset is required.

Parameter	Description
Web: CAS File EMS: Trunk Cas Table Index **[CASFileName_x]**	CAS file name (e.g., 'E_M_WinkTable.dat') that defines the CAS protocol (where x denotes the CAS file ID 0 to 7). It is possible to define up to eight different CAS files by repeating this parameter. Each CAS file can be associated with one or more of the device's trunks, using the parameter CASTableIndex or it can be associated per B-channel using the parameter CASChannelIndex. **Note:** For this parameter to take effect, a device reset is required.
Web: Dial Plan EMS: Dial Plan Name **[CasTrunkDialPlanName_x]**	The Dial Plan name (up to 11-character strings) that is used on a specific trunk (denoted by x).
Web: Dial Plan File EMS: Dial Plan File Name **[DialPlanFileName]**	The name (and path) of the Dial Plan file (defining dial plans). This file should be constructed using the DConvert utility (refer to the Product Reference Manual).
[UserInfoFileName]	The name (and path) of the file containing the User Information data.

6.20.2 Automatic Update Parameters

The automatic update of software and configuration files parameters are described in the table below.

Table 6-71: Automatic Update of Software and Configuration Files Parameters

Parameter	Description
General Automatic Update Parameters	
[AutoUpdateCmpFile]	Enables or disables the Automatic Update mechanism for the cmp file. ▪ **[0]** = The Automatic Update mechanism doesn't apply to the cmp file (default). ▪ **[1]** = The Automatic Update mechanism includes the cmp file. **Note:** For this parameter to take effect, a device reset is required.
[AutoUpdateFrequency]	Determines the number of minutes the device waits between automatic updates. The default value is 0 (i.e., the update at fixed intervals mechanism is disabled). **Note:** For this parameter to take effect, a device reset is required.
[AutoUpdatePredefinedTime]	Schedules an automatic update to a user-defined time of the day. The format of this parameter is: 'HH:MM', where *HH* depicts the hour and *MM* the minutes, for example, 20:18. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ The actual update time is randomized by five minutes to reduce the load on the Web servers.

Parameter	Description
[ResetNow]	Invokes an immediate device reset. This option can be used to activate offline (i.e., not on-the-fly) parameters that are loaded using the parameter IniFileUrl. ▪ **[0]** = The immediate restart mechanism is disabled (default). ▪ **[1]** = The device immediately resets after an *ini* file with this parameter set to 1 is loaded.
Software/Configuration File URL Path for Automatic Update Parameters	
[CmpFileURL]	Specifies the name of the *cmp* file and the path to the server (IP address or FQDN) from where the device loads a new *cmp* file and updates itself. The *cmp* file can be loaded using HTTP/HTTPS, FTP, FTPS, or NFS. For example: http://192.168.0.1/filename **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ When this parameter is configured, the device always loads the *cmp* file after it is reset. ▪ The *cmp* file is validated before it's burned to flash. The checksum of the *cmp* file is also compared to the previously burnt checksum to avoid unnecessary resets. ▪ The maximum length of the URL address is 255 characters.
[IniFileURL]	Specifies the name of the *ini* file and the path to the server (IP address or FQDN) on which it is located. The *ini* file can be loaded using HTTP/HTTPS, FTP, FTPS, or NFS. For example: http://192.168.0.1/filename http://192.8.77.13/config<MAC> https://<username>:<password>@<IP address>/<file name> **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ When using HTTP or HTTPS, the date and time of the *ini* file are validated. Only more recently dated *ini* files are loaded. ▪ The optional string '<MAC>' is replaced with the device's MAC address. Therefore, the device requests an *ini* file name that contains its MAC address. This option allows the loading of specific configurations for specific devices. ▪ The maximum length of the URL address is 99 characters.
[PrtFileURL]	Specifies the name of the Prerecorded Tones file and the path to the server (IP address or FQDN) on which it is located. For example: http://server_name/file, https://server_name/file. **Note:** The maximum length of the URL address is 99 characters.
[CptFileURL]	Specifies the name of the CPT file and the path to the server (IP address or FQDN) on which it is located. For example: http://server_name/file, https://server_name/file. **Note:** The maximum length of the URL address is 99 characters.

Parameter	Description
[VpFileURL]	Specifies the name of the Voice Prompts file and the path to the server (IP address or FQDN) on which it is located. For example: http://server_name/file, https://server_name/file. **Notes:** ▪ The maximum length of the URL address is 99 characters. ▪ This parameter is applicable only to Mediant 1000.
[CasFileURL]	Specifies the name of the CAS file and the path to the server (IP address or FQDN) on which it is located. For example: http://server_name/file, https://server_name/file. **Note:** The maximum length of the URL address is 99 characters.
[TLSRootFileUrl]	Specifies the name of the TLS trusted root certificate file and the URL from where it's downloaded. **Note:** For this parameter to take effect, a device reset is required.
[TLSCertFileUrl]	Specifies the name of the TLS certificate file and the URL from where it's downloaded. **Note:** For this parameter to take effect, a device reset is required.
[UserInfoFileURL]	Specifies the name of the User Information file and the path to the server (IP address or FQDN) on which it is located. For example: http://server_name/file, https://server_name/file **Note:** The maximum length of the URL address is 99 characters.

7 Restoring Factory Default Settings

The device provides you with the following methods for restoring the device's configuration to factory default settings:

- Using the CLI (refer to "Restoring Defaults using CLI" on page 471)
- Loading an empty *ini* file (refer to "Restoring Defaults using an *ini* File" on page 471)
- Using the hardware Reset button (refer to Restoring Defaults using Hardware Reset Button on page 471)

7.1 Restoring Defaults using CLI

The device can be restored to factory defaults using the CLI command **RestoreFactorySettings** (rfs), as described in the procedure below.

➢ **To restore factory default settings using CLI:**

1. Access the device's CLI:
   a. Connect the device's RS-232 port (refer to the *Installation Manual*) to COM1 or COM2 communication port on your PC.
   b. Establish serial communication with the device, using a serial communication program (such as HyperTerminal™) with the following communication port settings:
      - **Baud Rate:** 115,200 bps
      - **Data Bits:** 8
      - **Parity:** None
      - **Stop Bits:** 1
      - **Flow Control:** None
2. At the CLI prompt, enter the following command:

```
RestoreFactorySettings
```

7.2 Restoring Defaults using an *ini* File

You can restore the device's parameters to default settings while retaining its IP address and the Web interface's login user name and password. This is achieved by loading an empty *ini* file to the device. The loaded *ini* file must be empty (i.e., no parameters) or have only semicolons ";" preceding all lines. When a parameter is absent from a loaded *ini* file, the default value is assigned to that parameter (according to the *cmp* file loaded to the device) and saved to the non-volatile memory (thereby, overriding the value previously defined for that parameter).

7.3 Restoring Defaults using Hardware Reset Button

The device's hardware Reset button can be used to reset the device to default settings. For a detailed description, refer to the device's *Installation Manual*.

Reader's Notes

8 Auxiliary Configuration Files

This section describes the auxiliary files that can be loaded (in addition to the *ini* file) to the device:

- Call Progress Tones (refer to "Call Progress Tones File" on page 473
- Distinctive Ringing in the *ini* file (refer to Distinctive Ringing on page 476)
- Prerecorded Tones (refer to "Prerecorded Tones File" on page 478
- Voice Prompts (refer to Voice Prompts File on page 479)
- CAS (refer to CAS Files on page 480)
- Dial Plan (refer to Dial Plan File on page 480)
- User Information (refer to "User Information File" on page 482)

You can load these auxiliary files to the device using one of the following methods:

- Loading the files directly to the device using the device's Web interface (refer to "Loading Auxiliary Files" on page 200)
- Specifying the auxiliary file name in the *ini* file (refer to "Auxiliary and Configuration Files Parameters" on page 467) and then loading the *ini* file to the device

8.1 Call Progress Tones File

The Call Progress Tones (CPT) and Distinctive Ringing (applicable to analog interfaces) auxiliary file is comprised of two sections:

- The first section contains the definitions of the Call Progress Tones (levels and frequencies) that are detected/generated by the device.
- The second section contains the characteristics of the Distinctive Ringing signals that are generated by the device (refer to Distinctive Ringing on page 476).

You can use one of the supplied auxiliary files (*.dat file format) or create your own file. To create your own file, it's recommended to modify the supplied *usa_tone.ini* file (in any standard text editor) to suit your specific requirements, and then convert the modified *ini* file into binary format using the TrunkPack Downloadable Conversion Utility (DConvert). For a description on converting a CPT *ini* file into a binary *dat* file, refer to the *Product Reference Manual*.

Note: Only the *dat* file format can be loaded to the device.

You can create up to 32 different Call Progress Tones, each with frequency and format attributes. The frequency attribute can be single or dual-frequency (in the range of 300 to 1980 Hz) or an Amplitude Modulated (AM). Up to 64 different frequencies are supported. Only eight AM tones, in the range of 1 to 128 kHz, can be configured (the detection range is limited to 1 to 50 kHz). Note that when a tone is composed of a single frequency, the second frequency field must be set to zero.

The format attribute can be one of the following:

- **Continuous:** A steady non-interrupted sound (e.g., a dial tone). Only the 'First Signal On time' should be specified. All other on and off periods must be set to zero. In this case, the parameter specifies the detection period. For example, if it equals 300, the tone is detected after 3 seconds (300 x 10 msec). The minimum detection time is 100 msec.
- **Cadence:** A repeating sequence of on and off sounds. Up to four different sets of on/off periods can be specified.
- **Burst:** A single sound followed by silence. Only the 'First Signal On time' and 'First Signal Off time' should be specified. All other on and off periods must be set to zero. The burst tone is detected after the off time is completed.

You can specify several tones of the same type. These additional tones are used only for tone detection. Generation of a specific tone conforms to the first definition of the specific tone. For example, you can define an additional dial tone by appending the second dial tone's definition lines to the first tone definition in the *ini* file. The device reports dial tone detection if either of the two tones is detected.

The Call Progress Tones section of the *ini* file comprises the following segments:

- **[NUMBER OF CALL PROGRESS TONES]:** Contains the following key:
'Number of Call Progress Tones' defining the number of Call Progress Tones that are defined in the file.
- **[CALL PROGRESS TONE #X]:** containing the Xth tone definition, starting from 0 and not exceeding the number of Call Progress Tones less 1 defined in the first section (e.g., if 10 tones, then it is 0 to 9), using the following keys:
  - **Tone Type:** Call Progress Tone types:
    - **[1]** Dial Tone
    - **[2]** Ringback Tone
    - **[3]** Busy Tone
    - **[7]** Reorder Tone
    - **[8]** Confirmation Tone
    - **[9]** Call Waiting Tone - heard by the called party
    - **[15]** Stutter Dial Tone
    - **[16]** Off Hook Warning Tone
    - **[17]** Call Waiting Ringback Tone - heard by the calling party
    - **[18]** Comfort Tone
    - **[23]** Hold Tone
    - **[46]** Beep Tone
  - **Tone Modulation Type:** Amplitude Modulated (1) or regular (0)
  - **Tone Form:** The tone's format can be one of the following:
    - Continuous (1)
    - Cadence (2)
    - Burst (3)
  - **Low Freq [Hz]:** Frequency (in Hz) of the lower tone component in case of dual frequency tone, or the frequency of the tone in case of single tone. This is not relevant to AM tones.

- **High Freq [Hz:** Frequency (in Hz) of the higher tone component in case of dual frequency tone, or zero (0) in case of single tone (not relevant to AM tones).
- **Low Freq Level [-dBm]:** Generation level 0 dBm to -31 dBm in dBm (not relevant to AM tones).
- **High Freq Level:** Generation level of 0 to -31 dBm. The value should be set to 32 in the case of a single tone (not relevant to AM tones).
- **First Signal On Time [10 msec]:** 'Signal On' period (in 10 msec units) for the first cadence on-off cycle. For continuous tones, this parameter defines the detection period. For burst tones, it defines the tone's duration.
- **First Signal Off Time [10 msec]:** 'Signal Off' period (in 10 msec units) for the first cadence on-off cycle (for cadence tones). For burst tones, this parameter defines the off time required after the burst tone ends and the tone detection is reported. For continuous tones, this parameter is ignored.
- **Second Signal On Time [10 msec]:** 'Signal On' period (in 10 msec units) for the second cadence on-off cycle. Can be omitted if there isn't a second cadence.
- **Second Signal Off Time [10 msec]:** 'Signal Off' period (in 10 msec units) for the second cadence on-off cycle. Can be omitted if there isn't a second cadence.
- **Third Signal On Time [10 msec]:** 'Signal On' period (in 10 msec units) for the third cadence on-off cycle. Can be omitted if there isn't a third cadence.
- **Third Signal Off Time [10 msec]:** 'Signal Off' period (in 10 msec units) for the third cadence on-off cycle. Can be omitted if there isn't a third cadence.
- **Fourth Signal On Time [10 msec]:** 'Signal On' period (in 10 msec units) for the fourth cadence on-off cycle. Can be omitted if there isn't a fourth cadence.
- **Fourth Signal Off Time [10 msec]:** 'Signal Off' period (in 10 msec units) for the fourth cadence on-off cycle. Can be omitted if there isn't a fourth cadence.
- **Carrier Freq [Hz]:** Frequency of the carrier signal for AM tones.
- **Modulation Freq [Hz]:** Frequency of the modulated signal for AM tones (valid range from 1 to 128 Hz).
- **Signal Level [-dBm]:** Level of the tone for AM tones.
- **AM Factor [steps of 0.02]:** Amplitude modulation factor (valid range from 1 to 50). Recommended values from 10 to 25.

Notes:

- When the same frequency is used for a continuous tone and a cadence tone, the 'Signal On Time' parameter of the continuous tone must have a value that is greater than the 'Signal On Time' parameter of the cadence tone. Otherwise, the continuous tone is detected instead of the cadence tone.
- The tones frequency must differ by at least 40 Hz between defined tones.

For example, to configure the dial tone to 440 Hz only, enter the following text:

```
[NUMBER OF CALL PROGRESS TONES]
Number of Call Progress Tones=1
#Dial Tone
[CALL PROGRESS TONE #0]
Tone Type=1
Tone Form =1 (continuous)
Low Freq [Hz]=440
High Freq [Hz]=0
Low Freq Level [-dBm]=10 (-10 dBm)
High Freq Level [-dBm]=32 (use 32 only if a single tone is
required)
First Signal On Time [10msec]=300; the dial tone is detected after
3 sec
First Signal Off Time [10msec]=0
Second Signal On Time [10msec]=0
Second Signal Off Time [10msec]=0
```

8.1.1 Distinctive Ringing

Distinctive Ringing is applicable only to FXS interfaces. Using the Distinctive Ringing section of the Call Progress Tones auxiliary file, you can create up to 16 Distinctive Ringing patterns. Each ringing pattern configures the ringing tone frequency and up to four ringing cadences. The same ringing frequency is used for all the ringing pattern cadences. The ringing frequency can be configured in the range of 10 to 200 Hz with a 5 Hz resolution.

Each of the ringing pattern cadences is specified by the following parameters:

- **Burst Ring On Time:** Configures the cadence to be a burst cadence in the entire ringing pattern. The burst relates to On time and the Off time of the same cadence. It must appear between 'First/Second/Third/Fourth' string and the 'Ring On/Off Time' This cadence rings once during the ringing pattern. Otherwise, the cadence is interpreted as cyclic: it repeats for every ringing cycle.
- **Ring On Time:** Specifies the duration of the ringing signal.
- **Ring Off Time:** Specifies the silence period of the cadence.

The Distinctive Ringing section of the *ini* file format contains the following strings:

- **[NUMBER OF DISTINCTIVE RINGING PATTERNS]:** Contains the following key:
  - 'Number of Distinctive Ringing Patterns' defining the number of Distinctive Ringing signals that are defined in the file.
- **[Ringing Pattern #X]:** Contains the Xth ringing pattern definition (starting from 0 and not exceeding the number of Distinctive Ringing patterns defined in the first section minus 1) using the following keys:
  - **Ring Type:** Must be equal to the Ringing Pattern number.
  - **Freq [Hz]:** Frequency in hertz of the ringing tone.
  - **First (Burst) Ring On Time [10 msec]:** 'Ring On' period (in 10 msec units) for the first cadence on-off cycle.
  - **First (Burst) Ring Off Time [10 msec]:** 'Ring Off' period (in 10 msec units) for the first cadence on-off cycle.

- **Second (Burst) Ring On Time [10 msec]:** 'Ring On' period (in 10 msec units) for the second cadence on-off cycle.
- **Second (Burst) Ring Off Time [10 msec]:** 'Ring Off' period (in 10 msec units) for the second cadence on-off cycle.
- **Third (Burst) Ring On Time [10 msec]:** 'Ring On' period (in 10 msec units) for the third cadence on-off cycle.
- **Third (Burst) Ring Off Time [10 msec]:** 'Ring Off' period (in 10 msec units) for the third cadence on-off cycle.
- **Fourth (Burst) Ring On Time [10 msec]:** 'Ring Off' period (in 10 msec units) for the fourth cadence on-off cycle.
- **Fourth (Burst) Ring Off Time [10 msec]:** 'Ring Off' period (in 10 msec units) for the fourth cadence on-off cycle.

Note: In SIP, the Distinctive Ringing pattern is selected according to the Alert-Info header in the INVITE message. For example:
Alert-Info:<Bellcore-dr2>, or Alert-Info:<http://…/Bellcore-dr2>
'dr2' defines ringing pattern #2. If the Alert-Info header is missing, the default ringing tone (0) is played.

An example of a **ringing burst** definition is shown below:

```
#Three ringing bursts followed by repeated ringing of 1 sec on and
3 sec off.
[NUMBER OF DISTINCTIVE RINGING PATTERNS]
Number of Ringing Patterns=1

[Ringing Pattern #0]
Ring Type=0
Freq [Hz]=25
First Burst Ring On Time [10msec]=30
First Burst Ring Off Time [10msec]=30
Second Burst Ring On Time [10msec]=30
Second Burst Ring Off Time [10msec]=30
Third Burst Ring On Time [10msec]=30
Third Burst Ring Off Time [10msec]=30
Fourth Ring On Time [10msec]=100
Fourth Ring Off Time [10msec]=300
```

An example of various ringing signals definition is shown below:

```
[NUMBER OF DISTINCTIVE RINGING PATTERNS]
Number of Ringing Patterns=3

#Regular North American Ringing Pattern
[Ringing Pattern #0]
Ring Type=0
Freq [Hz]=20
First Ring On Time [10msec]=200
First Ring Off Time [10msec]=400


#GR-506-CORE Ringing Pattern 1
[Ringing Pattern #1]
Ring Type=1
Freq [Hz]=20
First Ring On Time [10msec]=200
First Ring Off Time [10msec]=400
```

```
#GR-506-CORE Ringing Pattern 2
[Ringing Pattern #2]
Ring Type=2
Freq [Hz]=20
First Ring On Time [10msec]=80
First Ring Off Time [10msec]=40
Second Ring On Time [10msec]=80
Second Ring Off Time [10msec]=400
```

8.1.2 FXS Distinctive Ringing and Call Waiting Tones per Source Number

The device supports the configuration of a Distinctive Ringing tone and Call Waiting Tone per calling number for IP-to-Tel calls. This feature can be configured per FXS endpoint or for a range of FXS endpoints. Therefore, different tones can be played per FXS endpoint/s depending on the source number of the received call. This configuration is performed using the ToneIndex *ini* file table parameter, which maps Ringing and/or Call Waiting tones to source number prefixes per FXS endpoint/s.

Typically, the Ringing and/or Call Waiting tone played is indicated in the SIP Alert-info header field of the received INVITE message. If this header is not present in the received INVITE, then this feature is used and the tone played is according to the settings in this table.

For example, to configure Distinctive Ringing and Call Waiting tones of Index #9 in the CPT file for FXS endpoints 1 to 4 when a call is received from a source number with prefix 2, configure the following in the *ini* file:

```
[ToneIndex]
FORMAT ToneIndex Index = ToneIndex FXSPort First,
ToneIndex FXSPort Last, ToneIndex SourcePrefix,
ToneIndex_PriorityIndex;
ToneIndex_Index 0 = 0, 3, 2, 1;
[\ToneIndex]
FirstCallWaitingToneID=8
```

Note that the Call Waiting tone index = priority index + FirstCallWaitingToneID(*). For example, if you want to select the Call Waiting tone defined in the CPT file at Index #9, then you can enter 1 as the priority index and the value 8 for FirstCallWaitingToneID. The summation of these values equals 9, i.e., index #9.

8.2 Prerecorded Tones File

The CPT file mechanism has several limitations such as a limited number of predefined tones and a limited number of frequency integrations in one tone. To overcome these limitations and provide tone generation capability that is more flexible, the Prerecorded Tones (PRT) file can be used. If a specific prerecorded tone exists in the PRT file, it takes precedence over the same tone that exists in the CPT file and is played instead of it.

Note: The PRT are used only for generation of tones. Detection of tones is performed according to the CPT file.

The PRT is a **.dat* file containing a set of prerecorded tones that can be played by the device. Up to 40 tones (totaling approximately 10 minutes) can be stored in a single PRT file on the device's flash memory. The prerecorded tones are prepared offline using standard recording utilities (such as CoolEdit™) and combined into a single file using the DConvert utility (refer to the *Product Reference Manual*).

The raw data files must be recorded with the following characteristics:

- **Coders:** G.711 A-law or G.711 µ-law
- **Rate:** 8 kHz
- **Resolution:** 8-bit
- **Channels:** mono

Once created, the PRT file can then be loaded to the device using AudioCodes' BootP/TFTP utility or the Web interface (refer to "Loading Auxiliary Files" on page 200).

The prerecorded tones are played repeatedly. This allows you to record only part of the tone and then play the tone for the full duration. For example, if a tone has a cadence of 2 seconds on and 4 seconds off, the recorded file should contain only these 6 seconds. The PRT module repeatedly plays this cadence for the configured duration. Similarly, a continuous tone can be played by repeating only part of it.

8.3 Voice Prompts File

The Voice Prompts (VP) file contains a set of voice prompts (or announcements) that can be played by the device during operation. The voice announcements are prepared offline using standard recording utilities and then combined into a single file using the DConvert utility. The VP file can then be loaded to the device using the BootP/TFTP utility (refer to the Product Reference Manual) or Web interface.

The VP file is a collection of raw voice recordings and/or *wav* files. These recordings can be prepared using standard utilities such as CoolEdit and Goldwave™.

The raw data files must be recorded with the following characteristics:

- **Coders:** Linear G.711 A-law or G.711 µ-law
- **Rate:** 8000 kHz
- **Resolution:** 8-bit
- **Channels:** mono

When the list of recorded files is converted to a single *voiceprompts.dat* file, every Voice Prompt is tagged with an ID number, starting with '1'. This ID is later used by the device to play the correct announcement. Up to 1,000 Voice Prompts can be defined. If the size of the combined VP file is less than 1 MB, it can be permanently stored on flash memory. Larger files (up to 10 MB) are stored in RAM, and should be loaded again (using BootP/TFTP utility) after the device is reset.

The device can be provided with a professionally recorded English (U.S.) VP file.

Note: Voice Prompts are applicable only to Mediant 1000.

➢ **To generate and load the VP file:**

1. Prepare one or more voice files using standard utilities.
2. Use the DConvert utility to generate the *voiceprompts.dat* file from the pre-recorded voice messages (refer to the *Product Reference Manual*).
3. Load the *voiceprompts.dat* file to the device using TFTP (refer to the Product Reference Manual) or the Web interface (refer to "Loading Auxiliary Files" on page 200).

8.4 CAS Files

The CAS Protocol auxiliary files contain the CAS Protocol definitions that are used for CAS-terminated trunks. You can use the supplied files or construct your own files. Up to eight files can be loaded to the device. Different files can be assigned to different trunks (CASTableIndex_x) and different CAS tables can be assigned to different B-channels (CASChannelIndex).

The CAS files can be loaded to the device using the Web interface or *ini* file (refer to "Loading Auxiliary Files" on page 200).

Note: All CAS files loaded together must belong to the same Trunk Type (i.e., either E1 or T1).

8.5 Dial Plan File

The Dial Plan file contains a list of up to eight dial plans, supporting a total of up to 8,000 user-defined, distinct prefixes (e.g. area codes, international telephone number patterns) for the PSTN to which the device is connected. The Dial Plan is used for the following:

- ISDN Overlap Dialing, FXS, and FXO collecting digit mode (Tel-to-IP calls): The file includes up to eight patterns (i.e., eight dial plans). These allow the device to know when digit collection ends, after which it starts sending all the collected (or dialed) digits (in the INVITE message). This also provides enhanced digit mapping.
- CAS E1 MF-CR2 (Tel-to-IP calls): Useful for E1 MF-CR2 variants that do not support I-15 terminating digits (e.g., in Brazil and Mexico). The Dial Plan file allows the device to detect end-of-dialing in such cases. The CasTrunkDialPlanName_x *ini* file parameter determines which dial plan (in the Dial Plan file) to use for a specific trunk.

Note: To use this Dial Plan, you must also use a special CAS *.dat file that supports this feature (contact your AudioCodes sales representative).

- Prefix tags (for IP-to-Tel routing): Provides enhanced routing rules based on Dial Plan prefix tags. For a detailed description, refer to Dial Plan Prefix Tags for IP-to-Tel Routing on page 490.

The Dial Plan file is first created using a text-based editor (such as Notepad) and saved with the file extension *.*ini*. This *ini* file is then converted to a binary file (*.dat) using the DConvert utility (refer to the *Product Reference Manual*). Once converted, it can then be loaded to the device using the Web interface (refer to "Loading Auxiliary Files" on page 200).

The Dial Plan file must be prepared in a textual *ini* file with the following syntax:

- Every line in the file defines a known dialing prefix and the number of digits expected to follow that prefix. The prefix must be separated from the number of additional digits by a comma (',').
- Empty lines are ignored.
- Lines beginning with a semicolon (';') are ignored.
- Multiple dial plans may be specified in one file; a name in square brackets on a separate line indicates the beginning of a new dial plan. Up to eight dial plans can be defined.
- Asterisks ('*') and number-signs ('#') can be specified as part of the prefix.
- Numeric ranges are allowed in the prefix.
- A numeric range is allowed in the number of additional digits.

Notes:

- The prefixes must not overlap. Attempting to process an overlapping configuration by the DConvert utility results in an error message specifying the problematic line.
- For a detailed description on working with Dial Plan files, refer to "External Dial Plan File" on page 488.

An example of a Dial Plan file in *ini*-file format (i.e., before converted to *.dat) that contains two dial plans is shown below:

```
; Example of dial-plan configuration.
; This file contains two dial plans:

[ PLAN1 ]
; Defines cellular/VoIP area codes 052, 054, and 050.
; In these area codes, phone numbers have 8 digits.
052,8
054,8
050,8
; Defines International prefixes 00, 012, 014.
; The number following these prefixes may
; be 7 to 14 digits in length.

00,7-14
012,7-14
014,7-14
```

```
; Defines emergency number 911.
; No additional digits are expected.
911,0
[ PLAN2 ]
; Defines area codes 02, 03, 04.
; In these area codes, phone numbers have 7 digits.
0[2-4],7
; Operator services starting with a star: *41, *42, *43.
; No additional digits are expected.
*4[1-3],0
```

8.6 User Information File

The User Information file is a text file that maps PBX extensions connected to the device to global IP numbers. In this context, a global IP phone number (alphanumerical) serves as a routing identifier for calls in the 'IP world'. The PBX extension uses this mapping to emulate the behavior of an IP phone.

Note: By default, the mapping mechanism is disabled and must be activated using the parameter EnableUserInfoUsage.

The maximum size of the file is 10,800 bytes (for analog modules) and 108,000 bytes for digital modules. Each line in the file represents a mapping rule of a single PBX extension. Up to 1,000 rules can be configured. Each line includes five items separated with commas. The items are described in the table below:

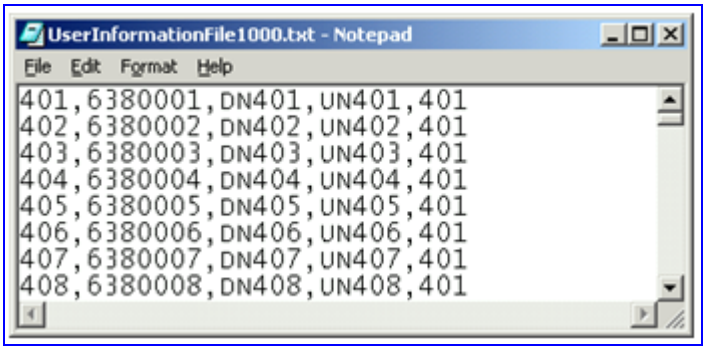
Table 8-1: User Information Items

Item	Description	Maximum Size (Characters)
PBX extension #	The relevant PBX extension number.	10
Global phone #	The relevant global phone number.	20
Display name	A string that represents the PBX extensions for the Caller ID.	30
Username	A string that represents the user name for SIP registration.	40
Password	A string that represents the password for SIP registration.	20

Note: For FXS ports, when the device is required to send a new request with the 'Authorization' header (for example, after receiving a SIP 401 reply), it uses the user name and password from the Authentication table. To use the username and password from the User Info file, change the parameter 'Password' from its default value.

An example of a User Information file is shown in the figure below:

Figure 8-1: Example of a User Information File



Note: The last line in the User Information file must end with a carriage return (i.e., by pressing the <Enter> key).

The User Information file can be loaded to the device by using one of the following methods:

- *ini* file, using the parameter UserInfoFileName (described in "Auxiliary and Configuration Files Parameters" on page 467)
- Web interface (refer to "Loading Auxiliary Files" on page 200)
- Automatic update mechanism, using the parameter UserInfoFileURL (refer to the *Product Reference Manual*)

Each PBX extension registers separately (a REGISTER message is sent for each entry only if AuthenticationMode is set to Per Endpoint) using the IP number in the From/To headers. The REGISTER messages are sent gradually. Initially, the device sends requests according to the maximum number of allowed SIP dialogs (configured by the parameter NumberOfActiveDialogs). After each received response, the subsequent request is sent. Therefore, no more than NumberOfActiveDialogs dialogs are active simultaneously. The user name and password are used for SIP Authentication when required.

The calling number of outgoing Tel-to-IP calls is first translated to an IP number and then (if defined), the manipulation rules are performed. The Display Name is used in the From header in addition to the IP number. The called number of incoming IP-to-Tel calls is translated to a PBX extension only after manipulation rules (if defined) are performed.

Reader's Notes

9 IP Telephony Capabilities

This section describes the device's main IP telephony capabilities.

9.1 Dialing Plan Features

This section discusses various dialing plan features supported by the device:

- Dialing plan notations (refer to "Dialing Plan Notation for Routing and Manipulation" on page 485)
- Digit mapping (refer to "Digit Mapping" on page 487)
- External Dial Plan file containing dial plans (refer to "External Dial Plan File" on page 488)
- Dial plan prefix tags for enhanced IP-to-Tel routing (refer to Dial Plan Prefix Tags for IP-to-Tel Routing on page 490)

9.1.1 Dialing Plan Notation for Routing and Manipulation

The device supports flexible dialing plan notations for representing digits (single or multiple) entered for destination and source prefixes (of phone numbers and SIP URI user names) in the routing tables.

Table 9-1: Dialing Plan Notations

Notation	Description	Example
[n-m]	Represents a range of numbers. **Note:** Range of letters is not supported.	▪ **[5551200-5551300]#:** represents all numbers from 5551200 to 5551300. ▪ **123[100-200]:** represents all numbers from 123100 to 123200.
[n,m,...]	Represents multiple numbers. Up to three digits can be used to denote each number.	▪ **[2,3,4,5,6]#:** represents a one-digit number starting with 2, 3, 4, 5, or 6. ▪ **[11,22,33]xxx#:** represents a five-digit number that starts with 11, 22, or 33. ▪ **[111,222]xxx#:** represents a six-digit number that starts with 111 or 222.
[n1-m1,n2-m2,a,b,c,n3-m3]	Represents a mixed notation of multiple ranges and single numbers. **Note:** The ranges and the single numbers must have the same number of digits. For example, each number range and single number in the dialing plan [123-130,455,577,780-790] consists of three digits.	**[123-130,455,766,780-790]:** represents numbers 123 to 130, 455, 766, and 780 to 790.

Notation	Description	Example
x	Represents any single digit.	-
Pound sign (#) at the end of a number	Represents the end of a number.	**54324xx#:** represents a 7-digit number that starts with 54324.
A single asterisk (*)	Represents any number.	***:** represents any number (i.e., all numbers).
x[n,l]y	For a description, refer to the text appearing after this table.	0[5,3]15

The device also supports a notation for adding a prefix where part of the prefix is first extracted from a user-defined location in the original destination or source number. This notation is entered in the 'Prefix to Add' field in the Number Manipulation tables:

x[n,l]y...

where,

- *x* = any number of characters/digits to add at the beginning of the number (i.e. first digits in the prefix).
- *[n,l]* = defines the location in the original destination or source number where the digits *y* are added:
  - *n* = location (number of digits counted from the left of the number) of a specific string in the original destination or source number.
  - *l* = number of digits that this string includes.
- y = prefix to add at the specified location.

For example, assume that you want to manipulate an incoming IP call with destination number +5492028888888 (area code 202 and phone number 8888888) to the number 0202158888888. To perform such a manipulation, the following configuration is required in the Number Manipulation table:

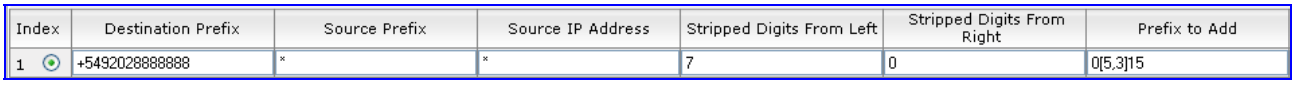
1. The following notation is used in the 'Prefix to Add' field:

   0[5,3]15

   where,

   - 0 is the number to add at the beginning of the original destination number.
   - [5,3] denotes a string that is located after (and including) the fifth character (i.e., the first '2' in the example) of the original destination number, and its length being three digits (i.e., the area code 202, in the example).
   - 15 is the number to add immediately after the string denoted by **[5,**3] - in other words, 15 is added after (i.e. to the right of) the digits 202.
2. The first seven digits from the left are removed from the original number, by entering "7" in the 'Stripped Digits From Left' field.

Figure 9-1: Prefix to Add Field with Notation



In this configuration, the following manipulation process occurs: 1) the prefix is calculated, 020215 in the example; 2) the first seven digits from the left are removed from the original number, in the example, the number is changed to 8888888; 3) the prefix that was previously calculated is then added.

9.1.2 Digit Mapping

Digit map pattern rules are used for Tel-to-IP ISDN overlap dialing (for digital interface). The device collects digits until a match is found in the user-defined digit pattern (e.g., for closed numbering schemes) or until a timer expires (e.g., for open numbering schemes). If a match is found or the timer expires, the digit collection process is terminated.

The maximum number (up to 49) of collected destination number digits that can be received (i.e., dialed) from the Tel side by the device can be defined (using the parameter MaxDigits). When the number of collected digits reaches the maximum (or a digit map pattern is matched), the device uses these digits for the called destination number.

Dialing ends (and the device starts sending the digits) when any of the following scenarios occur:

- Maximum number of digits is received.
- Inter-digit timeout expires (up to 10 seconds). This is defined by using the parameter TimeBetweenDigits. This is the time that the device waits between each received digit. When this inter-digit timeout expires, the device uses the collected digits to dial the called destination number.
- Pound (#) key is pressed.
- Digit map pattern is matched.

Digit map (pattern) rules are defined by the parameter DigitMapping. This is used to reduce the dialing period for ISDN Overlap dialing (ISDNRxOverlap is set to 1) for digital interfaces. If the digit string (i.e., dialed number) matches one of the patterns in the digit map, the device stops collecting digits and establishes a call with the collected number.

The digit map pattern can contain up to 52 options (rules), each separated by a vertical bar (|). The maximum length of the entire digit pattern is 152 characters. The available notations are described in the table below:

Table 9-2: Digit Map Pattern Notations

Notation	Description
[n-m]	Range of numbers (not letters).
.	(single dot) Repeat digits until next notation (e.g., T).
x	Any single digit.
T	Dial timeout (configured by the parameter TimeBetweenDigits).
S	Immediately applies a specific rule that is part of a general rule. For example, if a digit map includes a general rule 'x.T' and a specific rule '11x', for the specific rule to take precedence over the general rule, append 'S' to the specific rule (i.e., '11xS').

Below is an example of a digit map pattern containing eight rules:

```
DigitMapping = 11xS|00[1-
7]xxx|8xxxxxxx|#xxxxxxx|*xx|91xxxxxxxxxx|9011x|x.T
```

In the example above, the rule "00[1-7]xxx" denotes dialed numbers that begin with 00, and then any digit from 1 through 7, followed by three digits (of any number).

Notes:

- If you want the device to accept/dial any number, ensure that the digit map contains the rule "x.T"; otherwise, dialed numbers not represented in the digit map are rejected.
- If an external Dial Plan is implemented for dialing plans (refer to "External Dial Plan File" on page 488), then digit mapping configured by the parameter DigitMapping is ignored.

9.1.3 External Dial Plan File

The device allows you to select a specific Dial Plan (index) defined in an external Dial Plan file. This file is loaded to the device as a *.dat file (binary file), converted from an *ini* file using the DConvert utility. This file can include up to eight Dial Plans (Dial Plan indices). The required Dial Plan can be selected using the Dial Plan index, using the parameter DialPlanIndex. This parameter can use values 0 through 7, where 0 denotes PLAN1, 1 denotes PLAN2, and so on. The Dial Plan index can be configured globally or per Tel Profile. The Dial Plan file can include up to 8,000 dialing rules (lines).

The format of the Dial Plan index file is as follows:

- A name in square brackets ("[...]") on a separate line indicates the beginning of a new Dial Plan index.
- Every line under the Dial Plan index defines a dialing prefix and the number of digits expected to follow that prefix. The prefix is separated by a comma (",") from the number of additional digits.
- The prefix can include numerical ranges in the format [x-y], as well as multiple numerical ranges [n-m][x-y] (no comma between them).
- The prefix can include asterisks ("*") and number signs ("#").
- The number of additional digits can include a numerical range in the format x-y.
- Empty lines and lines beginning with a semicolon (";") are ignored.

An example of a Dial Plan file with indices (in *ini*-file format before conversion to binary *.dat) is shown below:

```
[ PLAN1 ]
; Area codes 02, 03, - phone numbers include 7 digits.
02,7
03,7
; Cellular/VoIP area codes 052, 054 - phone numbers include 8
digits.
052,8
054,8
; International prefixes 00, 012, 014 - number following
prefixes include 7 to 14 digits.
00,7-14
012,7-14
014,7-14
; Emergency number 911 (no additional digits expected).
911,0
[ PLAN2 ]
; Supplementary services such as Call Camping and Last Calls
(no additional digits expected), by dialing *41, *42, or *43.
*4[1-3],0
```

Notes:

- If the external Dial Plan file is used for digit mapping rules, then the parameter DigitMapping is ignored.
- For E1 CAS MFC-R2 variants (which don't support terminating digit for the called party number, usually I-15), the external Dial Plan file and the DigitMapping parameter are ignored. Instead, you can define a Dial Plan template per trunk using the parameter CasTrunkDialPlanName_x.

9.1.3.1 Modifying ISDN-to-IP Calling Party Number

The device can use the Dial Plan file to change the Calling Party Number value (source number) in the received ISDN incoming call when sending to IP. For this feature, the Dial Plan file supports the following syntax: <ISDN Calling Party Number>,0,<new calling number>

The Dial Plan file can also include a range for the source number, using the syntax **[x-**y].

Below is an example of such a configuration in the Dial Plan file:

```
[ PLAN1 ]
; specific received number changed to 04343434181.
0567811181,0,04343434181
; number range that changes to 04343434181.
056788118[2-4],0,04343434181
```

The device adds the newly manipulated calling number to the URI user part in the From header, and to the Contact header of the SIP INVITE sent to the IP side. For example, a received Calling Number Party of 0567811181 that is changed to 04343434181 (see Dial Plan file example above) is sent to the IP with a SIP INVITE as follows:

```
Via: SIP/2.0/UDP 211.192.160.214:5060;branch=z9hG4bK3157667347
From: <sip:04343434181@kt.co.kr:5060>;tag=de0004b1
To: sip:01066557573@kt.co.kr:5060
Call-ID: 585e60ec@211.192.160.214
CSeq: 1 INVITE
Contact:<sip:04343434181@211.192.160.214:5060;transport=udp>
```

Notes:

- Tel-to-IP routing is performed on the original source number if the parameter 'Tel to IP Routing Mode' is set to 'Route calls before manipulation'.
- Tel-to-IP routing is performed on the modified source number as defined in the Dial Plan file, if the parameter 'Tel To IP Routing Mode' is set to 'Route calls after manipulation'.
- Source number Tel-to-IP manipulation is performed on the modified source number as defined in the Dial Plan file.

9.1.4 Dial Plan Prefix Tags for IP-to-Tel Routing

The device supports the use of string labels (or "tags") in the external Dial Plan file for tagging incoming IP-to-Tel calls. The special “tag” is added as a prefix to the called party number, and then the 'Inbound IP Routing Table' uses this “tag” instead of the original prefix. Manipulation is then performed after routing in the Manipulation table, which strips the “tag” characters before sending the call to the endpoint.

This feature resolves the limitation of entries in the 'Inbound IP Routing Table' (IP-to-Tel call routing) for scenarios in which many different routing rules are required. For example, a city may have many different area codes, some for local calls and others for long distance calls (e.g. 425-202-xxxx for local calls, but 425-200-xxxx for long distance calls).

For using tags, the Dial Plan file is defined as follows:

- Number of dial plan (text)
- Dial string prefix (ranges can be defined in brackets)
- User-defined routing tag (text)

Note: Dial Plan Prefix Tags are not applicable to FXS and FXO interfaces.

The example configuration below assumes a scenario where multiple prefixes exist for local and long distance calls:

➢ **To use Dial Plan file routing tags:**

1. Load an *ini* file to the device that selects the Dial Plan index (e.g., 1) for routing tags, as shown below:

```
IP2TelTaggingDestDialPlanIndex = 1
```

2. Define the external Dial Plan file with two routing tags (as shown below):
   - "LOCL" - for local calls
   - "LONG" - for long distance calls

```
[ PLAN1 ]
42520[3-5],0,LOCL
425207,0,LOCL
42529,0,LOCL
425200,0,LONG
425100,0,LONG
```

Therefore, if an incoming IP call to destination prefix 425203 (for example) is received, the device adds the prefix tag "LOCL" (as specified in the Dial Plan file), resulting in the number "LOCL425203".

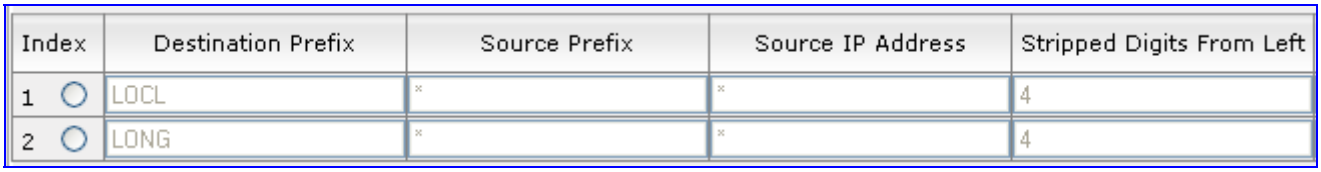
3. Assign the different tag prefixes to different Trunk Groups in the 'Inbound IP Routing Table':
   - The 'Dest. Phone Prefix' field is set to the value "LOCL" and this rule is assigned to a local Trunk Group (e.g. Trunk Group ID 1).
   - The 'Dest. Phone Prefix' field is set to the value "LONG" and this rule is assigned to a long distance Trunk Group (e.g. Trunk Group ID 2).

Figure 9-2: Configuring Dial Plan File Label for IP-to-Tel Routing

The above routing rules are configured to be performed before manipulation (described in the step below).

4. Configure manipulation in the 'Destination Phone Number Manipulation Table for IP to Tel Calls' table for removing the first four characters of the called party number “tag” (in our example, "LOCL" and "LONG"):
   - The 'Destination Prefix' field is set to the value "LOCL" and the 'Stripped Digits From Left' field is set to '4'.
   - The 'Destination Prefix' field is set to the value "LONG" and the 'Stripped Digits From Left' field is set to '4'.

Figure 9-3: Configuring Manipulation for Removing Label



9.2 Routing Applications

9.2.1 IP-to-IP Routing Application

The device's supports IP-to-IP VoIP call routing (or SIP Trunking). The IP-to-IP call routing application enables enterprises to seamlessly connect their IP-based PBX (IP-PBX) to SIP trunks, typically provided by an Internet Telephony Service Provider (ITSP). By implementing the device, enterprises can then communicate with PSTN networks (local and overseas) through ITSP's, which interface directly with the PSTN. Therefore, the IP-to-IP application enables enterprises to replace the bundles of physical PSTN wires with SIP trunks provided by ITSP's and use VoIP to communicate within and outside the enterprise network using its standard Internet connection. At the same time, the device can also provide an interface with the traditional PSTN network, enabling PSTN fallback in case of IP connection failure with the ITSP's.

In addition, the device supports multiple SIP Trunking. This can be useful in scenarios where if a connection to one ITSP fails, the call can immediately be transferred to another ITSP. In addition, by allowing multiple SIP trunks where each trunk is designated a specific ITSP, the device can route calls to an ITSP based on call destination (e.g., country code).

Therefore, in addition to providing VoIP communication within an enterprise's LAN, the device allows the enterprise to communicate outside of the corporate LAN using SIP Trunking. This includes remote (roaming) IP-PBX users, for example, employees using their laptops to communicate with one another from anywhere in the world such as at airports.

The IP-to-IP application can be implemented by enterprises in the following example scenarios:

- VoIP between an enterprise's headquarters and remote branch offices
- VoIP between an enterprise and the PSTN via an ITSP.

The IP-to-IP call routing capability is feature-rich, allowing interoperability with different ITSP's or service providers:

- Easy and smooth integration with multiple ITSP SIP trunks.
- Supports SIP registration and authentication with ITSP servers (on behalf of the enterprise's IP telephony system) even if the enterprise's IP telephony system does no support registration and authentication.
- Supports SIP-over-UDP, SIP-over-TCP, and SIP-over-TLS transport protocols, one of which is generally required by the ITSP.
- Provides alternative routing to different destinations (to another ITSP or the PSTN) when the connection with an ITSP network is down.
- Provides fallback to the legacy PSTN telephone network upon Internet connection failure.
- Provides Transcoding from G.711 to G.729 coder with the ITSP for bandwidth reduction.
- Supports SRTP, providing voice traffic security toward the ITSP.
- IP-to-IP routing can be used in combination with the regular Gateway application. For example, an incoming IP call can be sent to an E1/T1 span or it can be forwarded to an IP destination.

Therefore, the device provides the ideal interface between enterprises' IP-PBX's and ITSP SIP trunks. To facilitate the understanding of the IP-to-IP feature, this section provides a configuration example.

9.2.1.1 Theory of Operation

The device's IP-to-IP SIP session is performed by implementing Back-to-Back User Agent (B2BUA). The device acts as a user agent for both ends (*legs*) of the SIP call (from call establishment to termination). The session negotiation is performed independently for each call leg, using global parameters such as coders or using IP Profiles associated with each call leg to assign different configuration behaviors for these two IP-to-IP call legs.

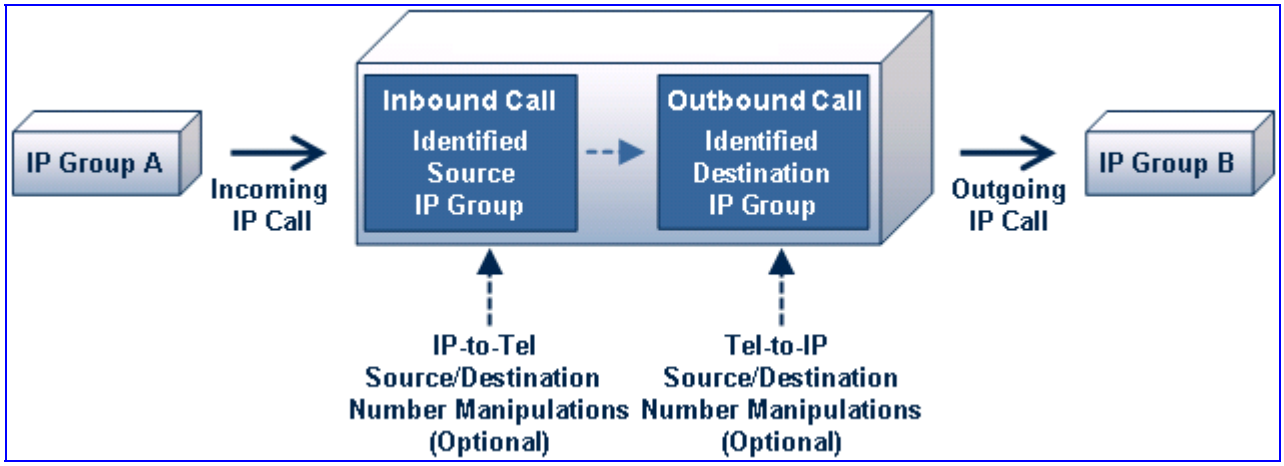
Currently, if transcoding is needed, the RTP streams for IP-to-IP calls transverse through the device and two DSP channels are allocated per IP-to-IP session. Therefore, the maximum number of media channels that can be designated for IP-to-IP call routing is 120 (corresponding to 60 IP-to-IP sessions). If transcoding is not needed, the device can support up to 150 IP-to-IP SIP sessions (without using DSP channels).

RTP-to-SRTP interworking requires one DSP channel. Therefore, the device supports up to 120 RTP-to-SRTP SIP sessions.

The device also supports NAT traversal for the SIP clients that are behind NAT. In this case, the device is defined with a global IP address.

The figure below provides a simplified illustration of the device's handling of IP-to-IP call routing:

Figure 9-4: Basic Schema of the Device's IP-to-IP Call Handling



The basic IP-to-IP call handling process can be summarized as follows:

1. Incoming IP calls are identified as belonging to a specific logical entity in the network (referred to as a *Source IP Group*), according to Inbound IP Routing rules.
2. The Source IP Group is associated with a specific IP Group (*Destination IP Group*), and then sent to the appropriate destination address (defined by a *Proxy Set*) associated with this Destination IP Group.
3. Number manipulation can be performed at both legs (inbound and outbound).

The following subsections discuss the main terms associated with the IP-to-IP call routing application.

9.2.1.1.1 Proxy Sets

A Proxy Set is a group containing up to five Proxy servers (for Proxy load balancing and redundancy), defined by IP address or fully qualified domain name (FQDN). The Proxy Set is assigned to IP Groups (of type SERVER only), representing the address of the IP Group to where the device sends the INVITE message (**destination** of the call). Typically, for IP-to-IP call routing, two Proxy Sets are defined for call destination – one for each leg (i.e., each IP Group) of the call (i.e., both directions).

9.2.1.1.2 IP Groups

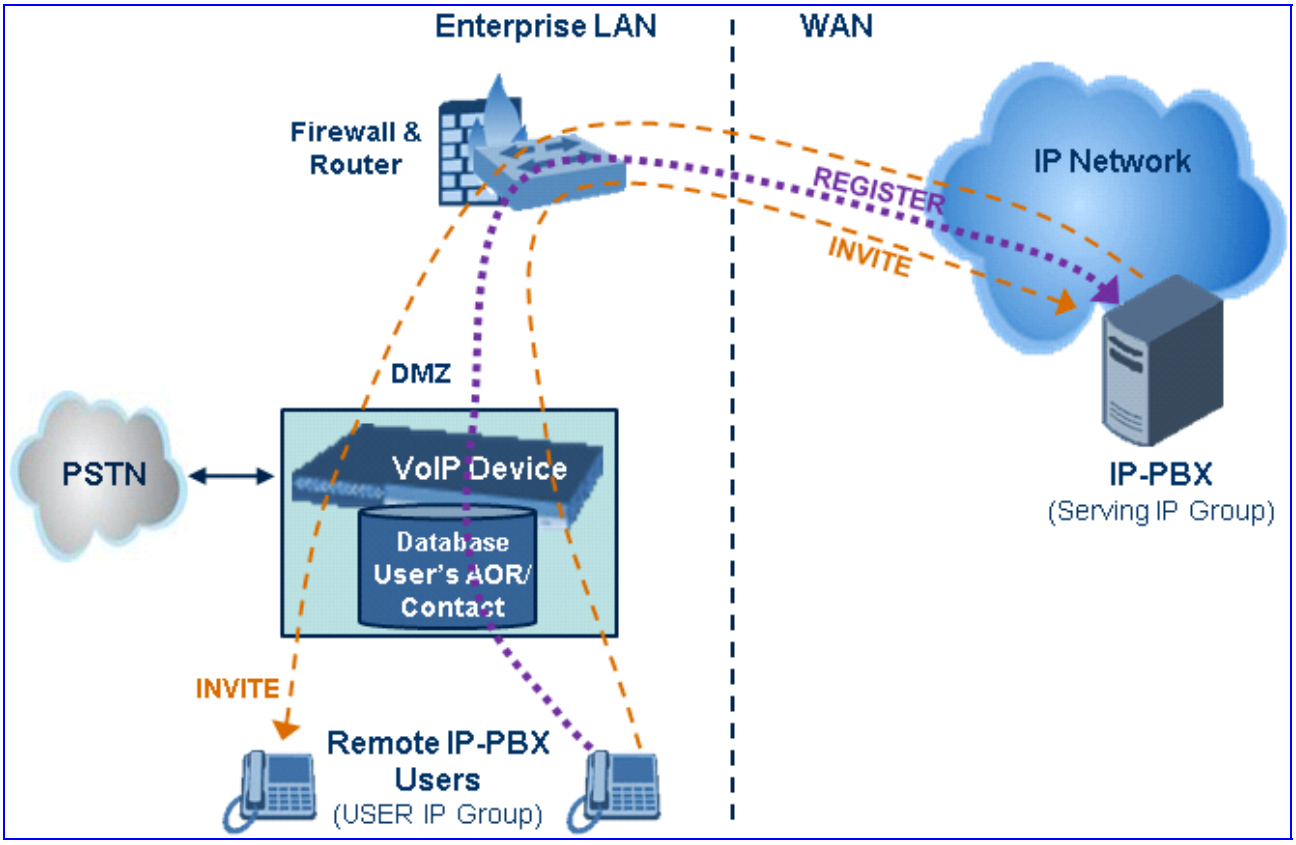
An IP Group represents a logical SIP entity in the device's network environment such as an ITSP SIP trunk, ITSP Proxy/Registrar server, IP-PBX, or remote IP-PBX users. The address of the IP Group is typically defined by the Proxy Set that is assigned to it.

The opposite legs of the call are each presented by an IP Group: one being a *Serving* IP Group; the other the *Served* IP Group. The Serving IP Group depicts the IP Group (e.g., ITSP) that provides service ("serves") to the Served IP Group (e.g., IP-PBX). This is the IP Group to where the device sends INVITE messages received from the Served IP Group as well as REGISTER messages for registering on behalf of the Served IP Group.

In addition, IP Groups can be *SERVER* or *USER* type. In SERVER IP Groups (e.g., ITSP or IP-PBX), the destination address (defined by the Proxy Set) is known. In contrast, USER IP Groups represents groups of users whose location is dynamically obtained by the device when REGISTER requests and responses traverse (or are terminated) by the device. Generally, these are remote IP-PBX users (e.g., IP phones and soft phones).

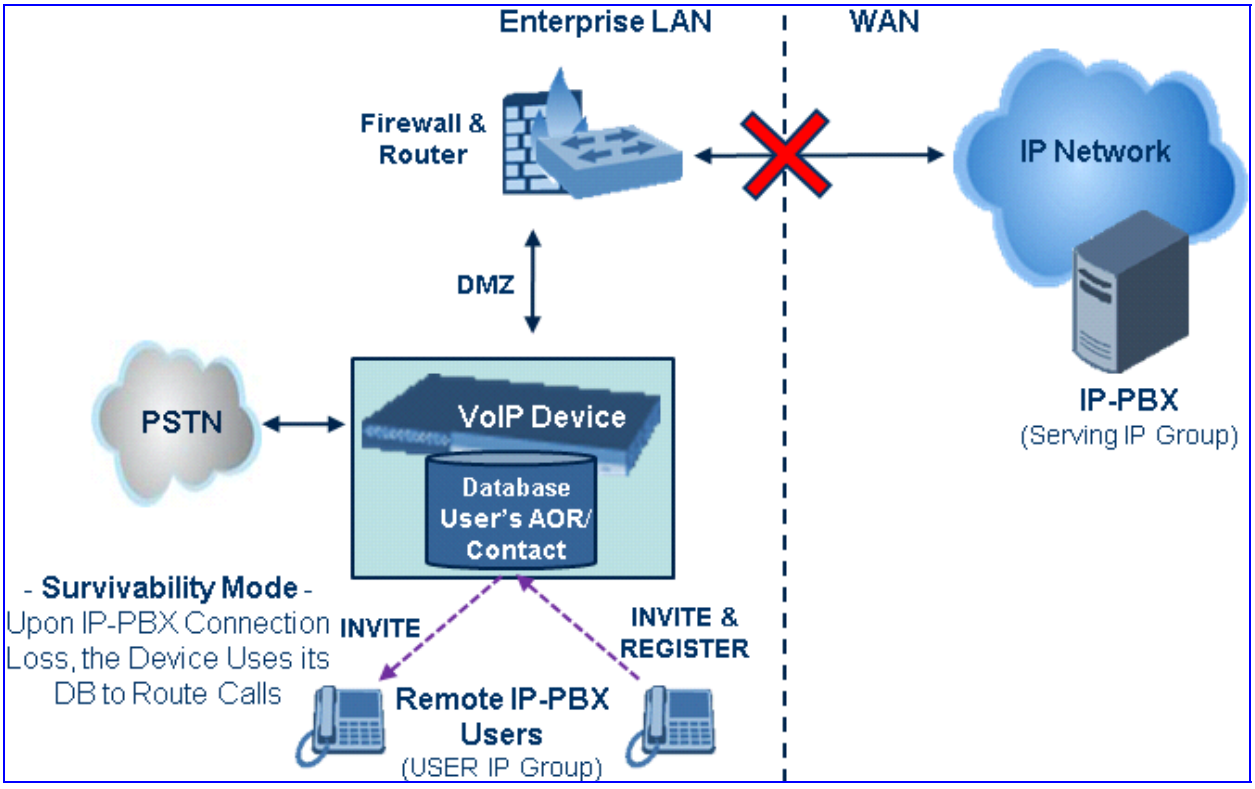
For registrations of USER IP Groups, the device updates its internal database with the AOR and Contacts of the users (refer to the figure below) Digest authentication using SIP 401/407 responses (if needed) is performed by the Serving IP Group (e.g., IP-PBX). The device forwards these responses directly to the remote SIP users. For a call to a registered remote user, the device searches its dynamic database (by using the Request URI) for an entry that matches a registered AOR or Contact. Once an entry is found, the IP destination is obtained and a SIP request is then sent to this user.

Figure 9-5: IP-to-IP Routing/Registration/Authentication of Remote IP-PBX Users (Example)



The device also supports the IP-to-IP call routing Survivability mode feature (refer to the figure below) for USER IP Groups. The device records (in its database) REGISTER messages sent by the clients of the USER IP Group. If communication with the Serving IP Group (e.g., IP-PBX) fails, the USER IP Group enters into Survivability mode in which the device uses its database for routing calls between the clients of the USER IP Group. The RTP packets between the clients traverse through the device. When the Serving IP Group is available again, the device returns to normal mode, sending INVITE and REGISTER messages to the Serving IP Group.

Figure 9-6: IP-to-IP Routing for IP-PBX Remote Users in Survivability Mode (Example)



9.2.1.1.3 Inbound and Outbound IP Routing Rules

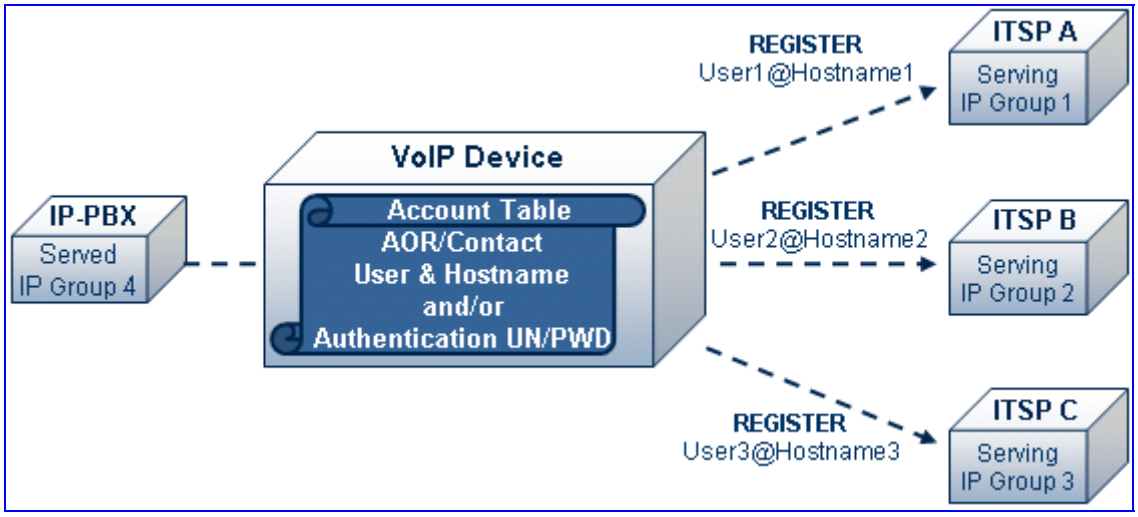
The device's IP-to-IP call routing is performed using the following two routing rule stages:

1. **Inbound IP Routing Mapping Rule:** Identifies the received call as an IP-to-IP call based on various characteristics such as the call's source IP address, and assigns it to an IP Group.
2. **Outbound IP Routing Mapping Rule:** Determines the destination (i.e., IP address) to where the incoming call (classified to a specific IP Group by the Inbound IP Routing rules) is finally routed. The destination address is typically depicted by another IP Group (destination IP Group) and therefore, the call is sent to the IP address that is defined in the Proxy Set associated with this IP Group. If the destination is a USER IP Group, the device searches for a match between the request URI (of the received INVITE) to an AOR registration record in the device's internal database. If a match is found, the INVITE is sent to the IP address of the registered contact.

9.2.1.1.4 Accounts

Accounts are used by the device to register to a Serving IP Group (e.g., an ITSP) on behalf of a Served IP Group (e.g., IP-PBX). This is necessary for ITSP's that require registration to provide services. Accounts are also used for defining user name/password for digest authentication (with or without registration) if required by the ITSP. Multiple Accounts per Served IP Group can be configured for registration to more than one Serving IP Group (e.g., an IP-PBX that requires registering to multiple ITSP's).

Figure 9-7: Registration with Multiple ITSP's on Behalf of IP-PBX



9.2.1.2 Configuring IP-to-IP Routing

This section provides step-by-step procedures for configuring IP-to-IP call routing. These procedures are based on the setup example described below. In this example, the device serves as the communication interface between the enterprise's IP-PBX (located on the LAN) and the following network entities:

- ITSP SIP trunks (located on the WAN)
- Remote IP-PBX users (located on the WAN)
- Local PSTN network

Calls from the Enterprise are routed according to destination.

This example assumes the following:

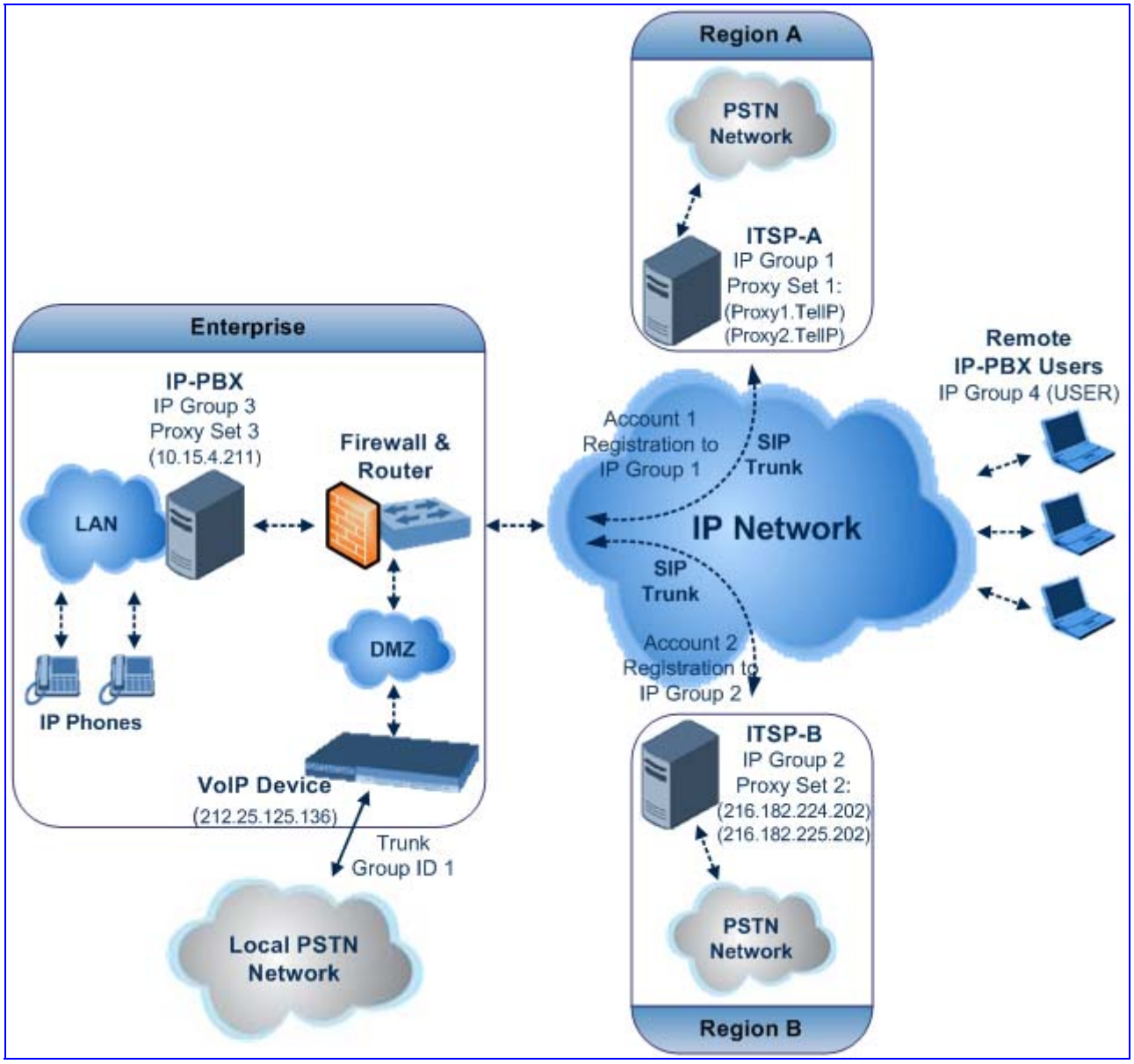
- The device has the public IP address 212.25.125.136 and is connected to the enterprise's firewall/NAT demilitarized zone (DMZ) network, providing the interface between the IP-PBX, and two ITSP's and the local PSTN.
- The enterprise has an IP-PBX located behind a Firewall/NAT:
  - IP-PBX IP address: 10.15.4.211
  - Transport protocol: UDP
  - Voice coder: G.711
  - IP-PBX users: 4-digit length extension number and served by two ITSPs.

- The enterprise also includes remote IP-PBX users that communicate with the IP-PBX via the device. All dialed calls from the IP-PBX consisting of four digits starting with digit "4" are routed to the remote IP-PBX users.

- Using SIP trunks, the IP-PBX connects (via the device) to two different ITSP's:
  - **ITSP-A:**
    - Implements Proxy servers with fully qualified domain names (FQDN): "Proxy1.ITSP-A" and "Proxy2.ITSP-B", using TLS.
    - Allocates a range of PSTN numbers beginning with +1919, which is assigned to a range of IP-PBX users.
    - Voice coder: G.723.
  - **ITSP-B:**
    - Implements Proxy servers with IP addresses 216.182.224.202 and 216.182.225.202, using TCP.
    - Allocates a range of PSTN numbers beginning with 0200, which is assigned to a range of IP-PBX users.
    - Voice coder: G.723.
- Registration and authentication is required by both ITSP's, which is performed by the device on behalf of the IP-PBX. The SIP REGISTER messages use different URI's (host name and contact user) in the From, To, and Contact headers per ITSP as well as username and password authentication.
- Outgoing calls from IP-PBX users are routed according to destination:
  - If the calls are dialed with the prefix "+81", they are routed to ITSP-A (Region A).
  - If the calls are dialed with the prefix "9", they are routed to the local PSTN network.
  - For all other destinations, the calls are routed to ITSP-B.
- The device is also connected to the PSTN through a traditional T1 ISDN trunk for local incoming and outgoing calls. Calls dialed from the enterprise's IP-PBX with prefix '9' are sent to the local PSTN. In addition, in case of Internet interruption and loss of connection with the ITSP trunks, all calls are rerouted to the PSTN.

The figure below provides an illustration of this example scenario:

Figure 9-8: SIP Trunking Setup Scenario Example



The steps for configuring the device according to the scenario above can be summarized as follows:

- Enable the IP-to-IP feature (refer to "Step 1: Enable the IP-to-IP Capabilities" on page 500).
- Configure the number of media channels (refer to "Step 2: Configure the Number of Media Channels" on page 500).
- Configure a Trunk Group for interfacing with the local PSTN (refer to "Step 3: Define a Trunk Group for the Local PSTN" on page 501).
- Configure Proxy Sets (refer to "Step 4: Configure the Proxy Sets" on page 502).
- Configure IP Groups (refer to "Step 5: Configure the IP Groups" on page 504).
- Configure Registration Accounts (refer to "Step 6: Configure the Account Table" on

page 508).

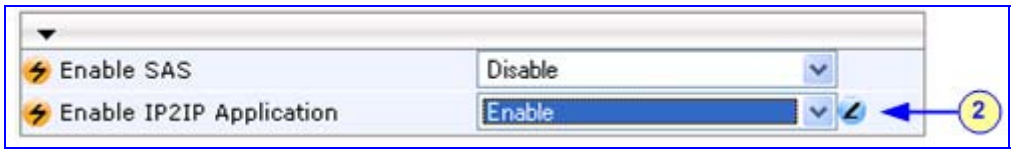
- Configure IP Profiles (refer to "Step 7: Configure IP Profiles for Voice Coders" on page 509).
- Configure inbound IP routing rules (refer to "Step 8: Configure Inbound IP Routing" on page 511).
- Configure outbound IP routing rules (refer to "Step 9: Configure Outbound IP Routing" on page 513).
- Configure destination phone number manipulation (refer to "Step 10: Configure Destination Phone Number Manipulation" on page 515).

9.2.1.2.1 Step 1: Enable the IP-to-IP Capabilities

This step describes how to enable the device's IP-to-IP application.

➢ **To enable IP-to-IP capabilities:**

1. Open the 'Applications Enabling' page (**Configuration** tab > **Protocol Configuration** menu > **Applications Enabling** page item).



2. From the 'Enable IP2IP Application' drop-down list, select "Enable".

Note: To enable the IP-to-IP application, the device must be loaded with the Software Upgrade Feature Key that includes the SBC feature.

9.2.1.2.2 Step 2: Configure the Number of Media Channels

The number of media channels represents the number of digital signaling processors (DSP) channels that the device allocates to IP-to-IP calls (the remaining DSP channels can be used for PSTN calls). Two IP media channels are used per IP-to-IP call. Therefore, the maximum number of media channels that can be designated for IP-to-IP call routing is 120 (corresponding to 60 IP-to-IP calls).

➢ **To configure the number of the media channels:**

1. Open the 'IP Media Settings' page (**Configuration** tab > **Media Settings** menu > **IP Media Settings** page item).

Figure 9-9: Defining Required Media Channels

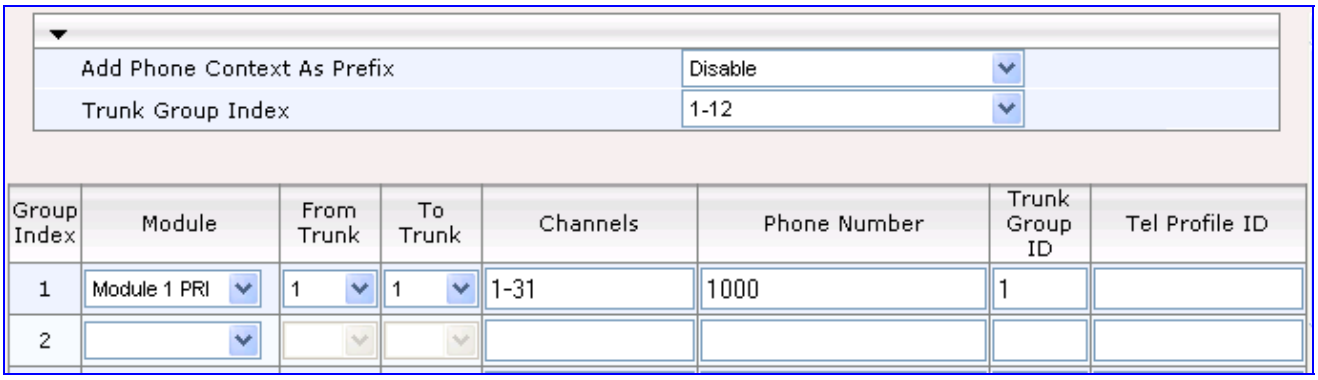
2. In the 'Number of Media Channels' field, enter the required number of media channels (in the example above, "120" to enable up to 60 IP-to-IP calls).
3. Click **Submit**.

9.2.1.2.3 Step 3: Define a Trunk Group for the Local PSTN

For incoming and outgoing local PSTN calls with the IP-PBX, you need to define the Trunk Group ID (#1) for the T1 ISDN trunk connecting between the device and the local PSTN. This Trunk Group is also used for alternative routing to the legacy PSTN network in case of a loss of connection with the ITSP's.

➢ **To configure the Trunk Group for local PSTN:**

1. Open the 'Trunk Group Table' page (**Protocol Configuration** menu > **Trunk Group** submenu > **Trunk Group** page item).
2. Configure Trunk Group ID #1 (as shown in the figure below):
   - From the 'From Trunk' and 'To Trunk' drop-down lists, select '1' to indicate Trunk 1 for this Trunk Group.
   - In the 'Channels' field, enter the Trunk channels or ports assigned to the Trunk Group (e.g. 1-31 for E1 and 1-24 for T1).
   - In the 'Phone Number' field, enter any phone number (logical) for this Trunk (e.g. 1000).
   - In the 'Trunk Group ID' field, enter '1' as the ID for this Trunk Group.



3. Configure the Trunk in the 'Trunk Settings' page (**PSTN Settings** menu > **Trunk Settings** page item).

9.2.1.2.4 Step 4: Configure the Proxy Sets

This step describes how to configure the following Proxy Sets:

- Proxy Set ID #1 defined with two FQDN's for ITSP-A.
- Proxy Set ID #2 defined with two IP addresses for ITSP-B.
- Proxy Set ID #3 defined with an IP address for the IP-PBX.

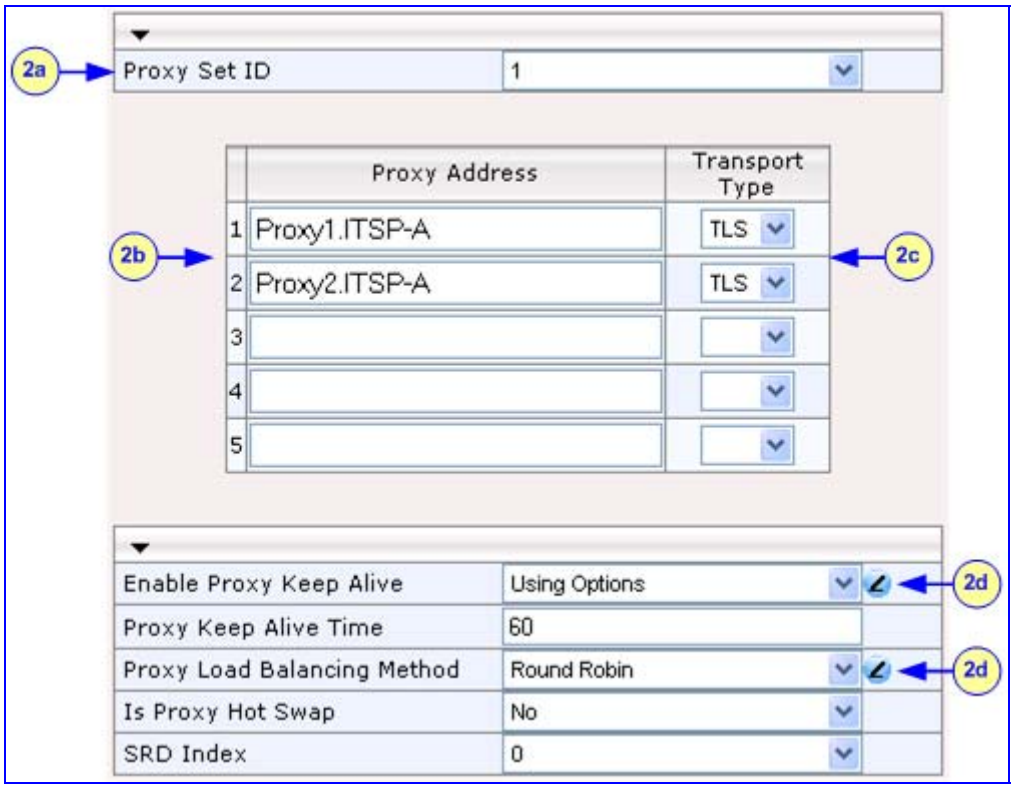
These Proxy Sets are later assigned to IP Groups (refer to "Step 5: Configure the IP Groups" on page 504).

Note that the Proxy Set represents the actual destination (IP address or FQDN) to which the call is routed.

➢ **To configure the Proxy Sets:**

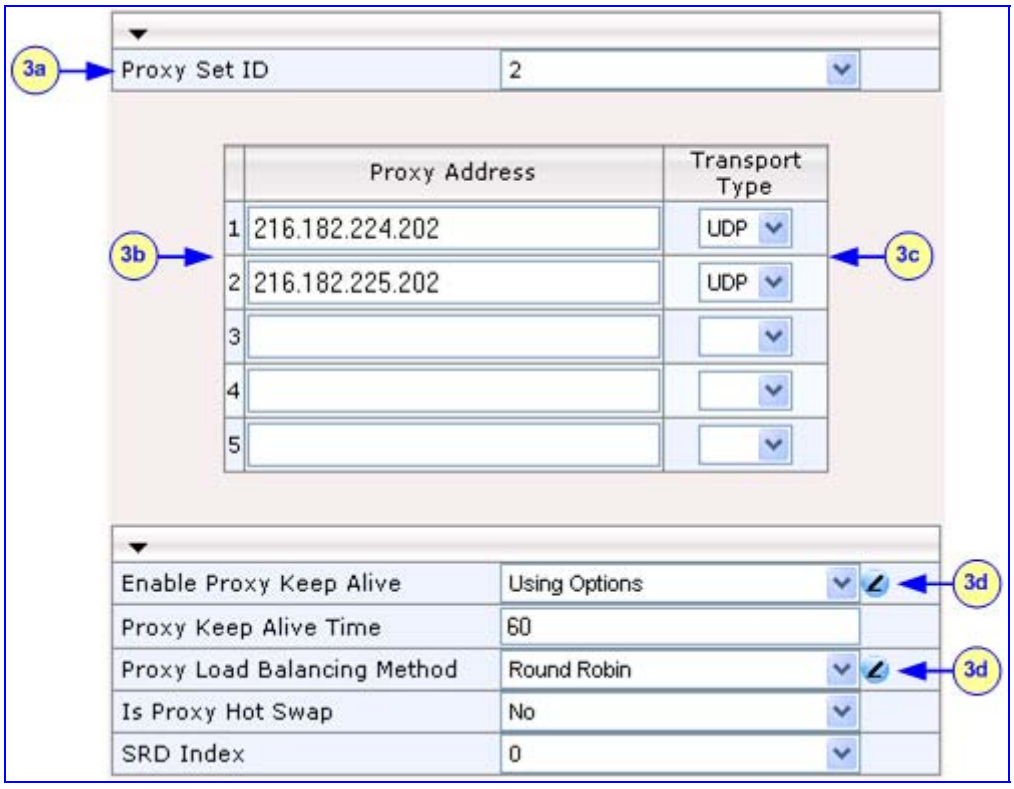
1. Open the 'Proxy Sets Table' page (**Protocol Configuration** menu > **Proxies, Registration, IP Groups** submenu > **Proxy Sets Table**).
2. Configure Proxy Set ID **#1** for ITSP-A:
   a. From the 'Proxy Set ID' drop-down list, select "1".
   b. In the 'Proxy Address' column, enter the FQDN of ITSP-A SIP trunk Proxy servers (e.g., Proxy1.ITSP-A and Proxy2. ITSP-A).
   c. From the 'Transport Type' drop-down list corresponding to the Proxy addresses entered above, select "TLS".
   d. In the 'Enable Proxy Keep Alive' drop-down list, select "Using Options", and then in the Proxy Load Balancing Method drop-down list, select "Round Robin".

Figure 9-10: Proxy Set ID #1 for ITSP-A



3. Configure Proxy Set ID **#2** for ITSP-B:
   a. From the 'Proxy Set ID' drop-down list, select "2".
   b. In the 'Proxy Address' column, enter the IP addresses of the ITSP-B SIP trunk (e.g., 216.182.224.202 and 216.182.225.202).
   c. From the 'Transport Type' drop-down list corresponding to the IP address entered above, select "UDP".
   d. In the 'Enable Proxy Keep Alive' drop-down list, select "Using Options", and then in the Proxy Load Balancing Method drop-down list, select "Round Robin".

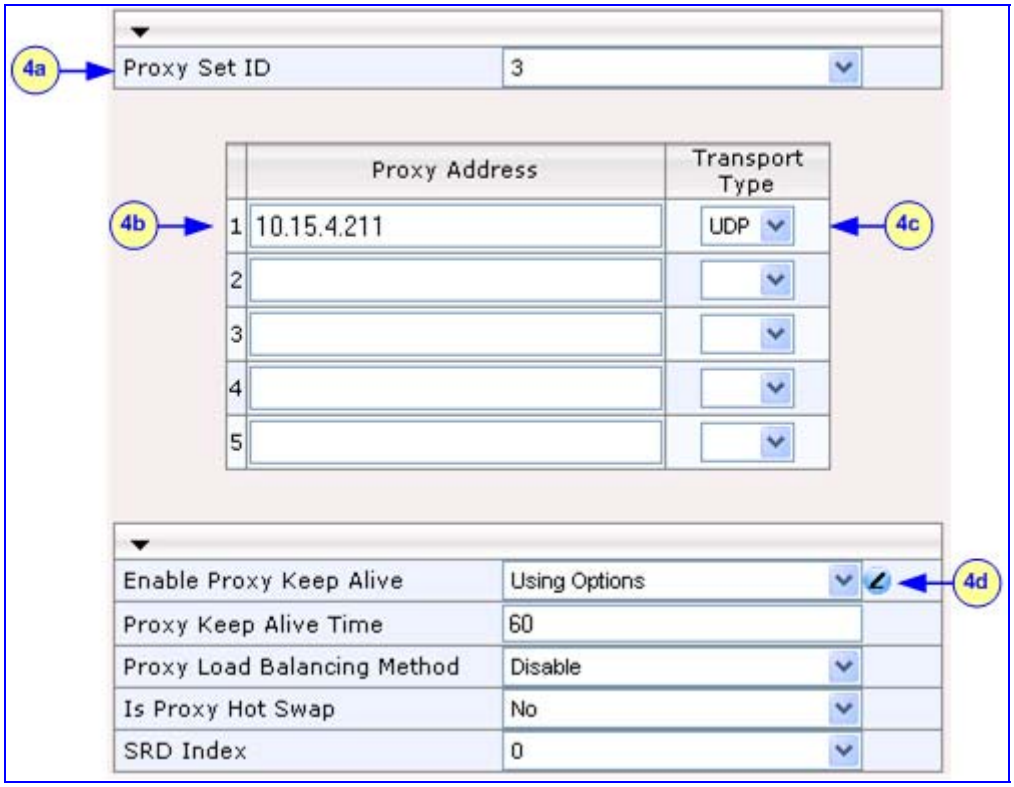
Figure 9-11: Proxy Set ID #2 for ITSP-B



4. Configure Proxy Set ID #3 for the IP-PBX:
   a. From the 'Proxy Set ID' drop-down list, select "3".
   b. In the 'Proxy Address' column, enter the IP address of the IP-PBX (e.g., 10.15.4.211).
   c. From the 'Transport Type' drop-down list corresponding to the IP address entered above, select "UDP ".

d. In the 'Enable Proxy Keep Alive' drop-down list, select "Using Options" – this is used in Survivability mode for remote IP-PBX users.

Figure 9-12: Proxy Set ID #3 for the IP-PBX



9.2.1.2.5 Step 5: Configure the IP Groups

This step describes how to create the IP Groups for the following entities in the network:

- ITSP-A SIP trunk
- ITSP-B SIP trunk
- IP-PBX
- IP-PBX remote users

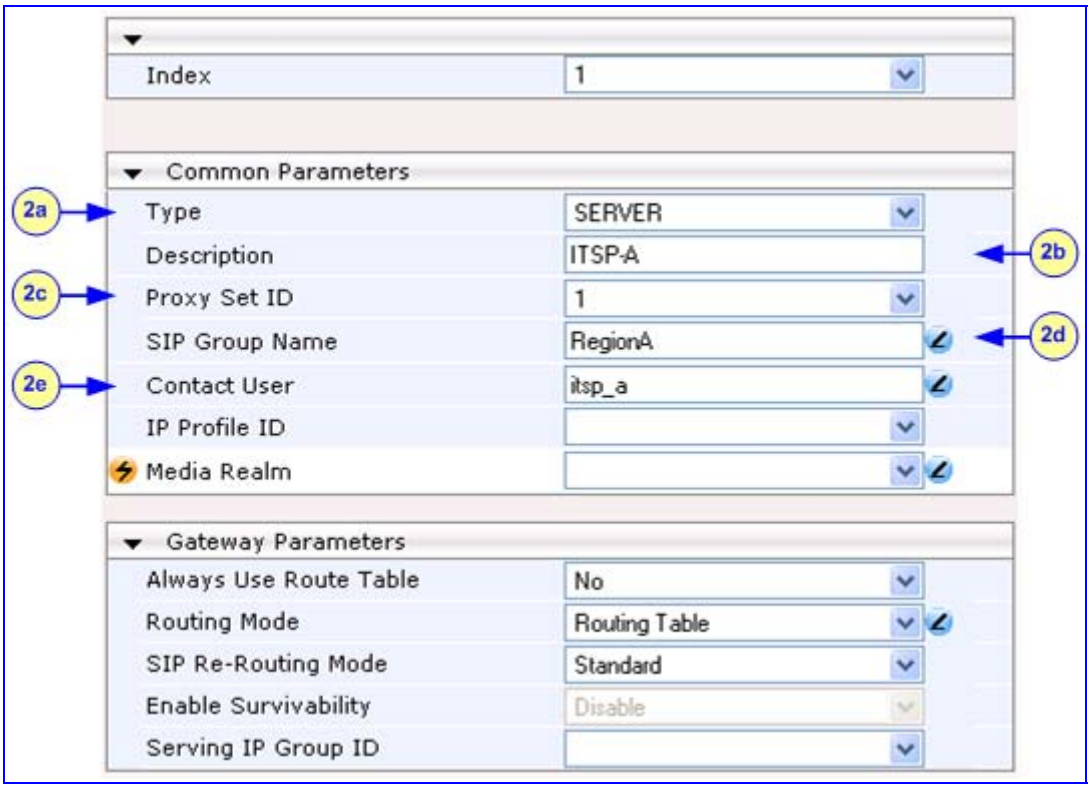
These IP Groups are later used by the device for routing calls.

➢ **To configure the IP Groups:**

1. Open the 'IP Group Table' page (**Protocol Configuration** menu > **Proxies, Registration, IP Groups** submenu > **IP Group Table**).
2. Define IP Group **#1** for ITSP-A:
   a. From the 'Type' drop-down list, select 'SERVER'.
   b. In the 'Description' field, type an arbitrary name for the IP Group (e.g., ITSP A).
   c. From the 'Proxy Set ID' drop-down lists, select '1' (represents the IP addresses, configured in , for communicating with this IP Group).
   d. In the 'SIP Group Name' field, enter the host name sent in the SIP Request From\To headers for this IP Group, as required by ITSP-A (e.g., RegionA).

e. Contact User = name that is sent in the SIP Request's Contact header for this IP Group (e.g., ITSP-A).

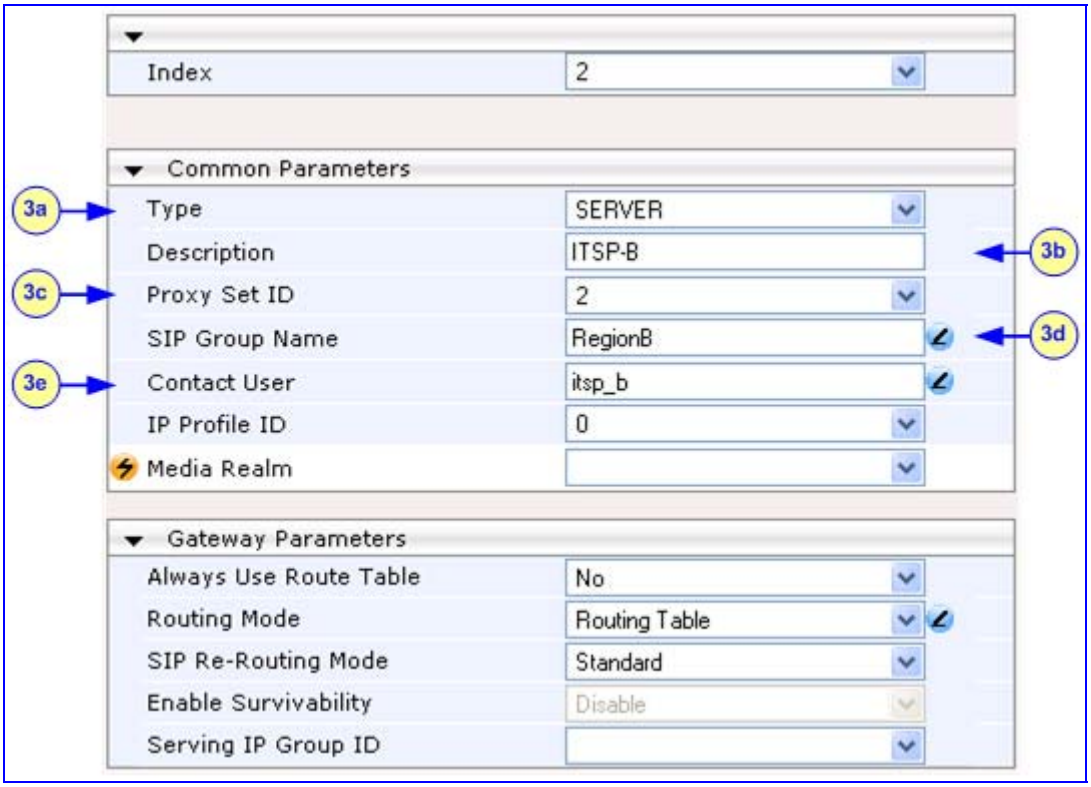
Figure 9-13: Defining IP Group 1



3. Define IP Group **#2** for ITSP-B:
   a. From the 'Type' drop-down list, select 'SERVER'.
   b. In the 'Description' field, type an arbitrary name for the IP Group (e.g., ITSP B).
   c. From the 'Proxy Set ID' drop-down lists, select '2' (represents the IP addresses, configured in **,** for communicating with this IP Group).
   d. In the 'SIP Group Name' field, enter the host name sent in SIP Request From\To headers for this IP Group, as required by ITSP-B (e.g., RegionB).

e. Contact User = name that is sent in the SIP Request Contact header for this IP Group (e.g., ITSP-B).

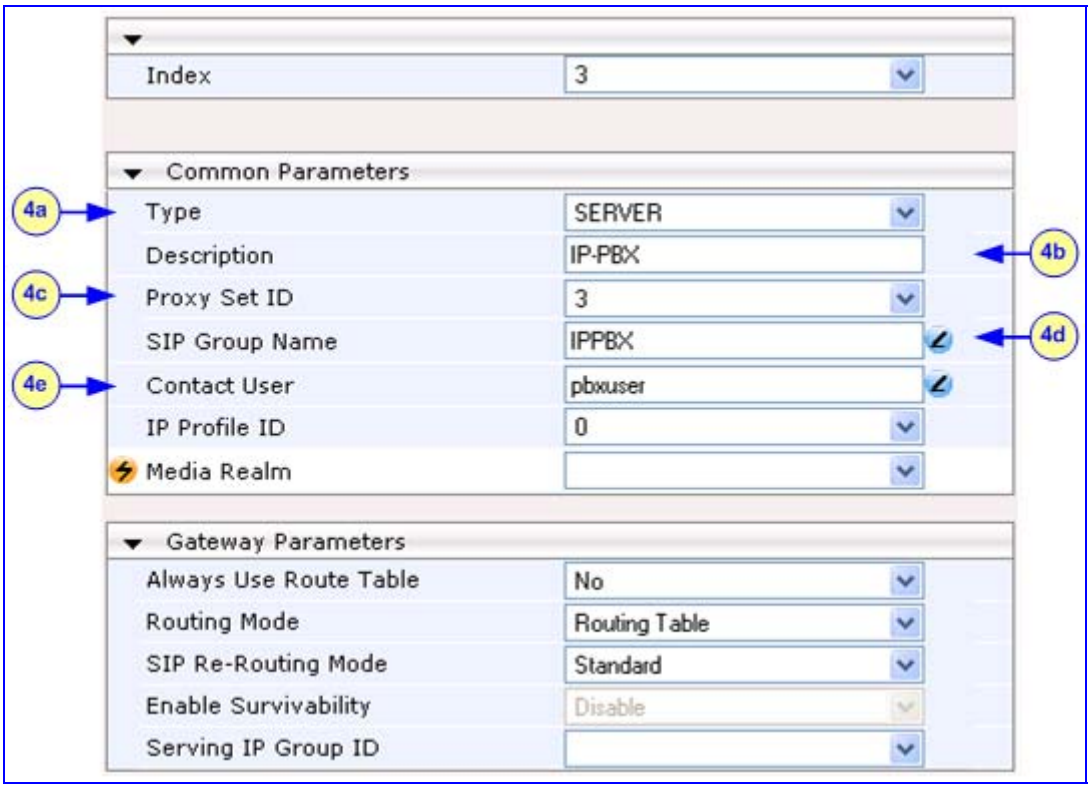
Figure 9-14: Defining IP Group 2



4. Define IP Group **#3** for the IP-PBX:
   a. From the 'Type' drop-down list, select 'SERVER'.
   b. In the 'Description' field, type an arbitrary name for the IP Group (e.g., IP-PBX).
   c. From the 'Proxy Set ID' drop-down lists, select '3' (represents the IP address, configured in , for communicating with this IP Group).
   d. In the 'SIP Group Name' field, enter the host name that is sent in SIP Request From\To headers for this IP Group (e.g., IPPBX).

e. Contact User = name that is sent in the SIP Request Contact header for this IP Group (e.g., PBXUSER).

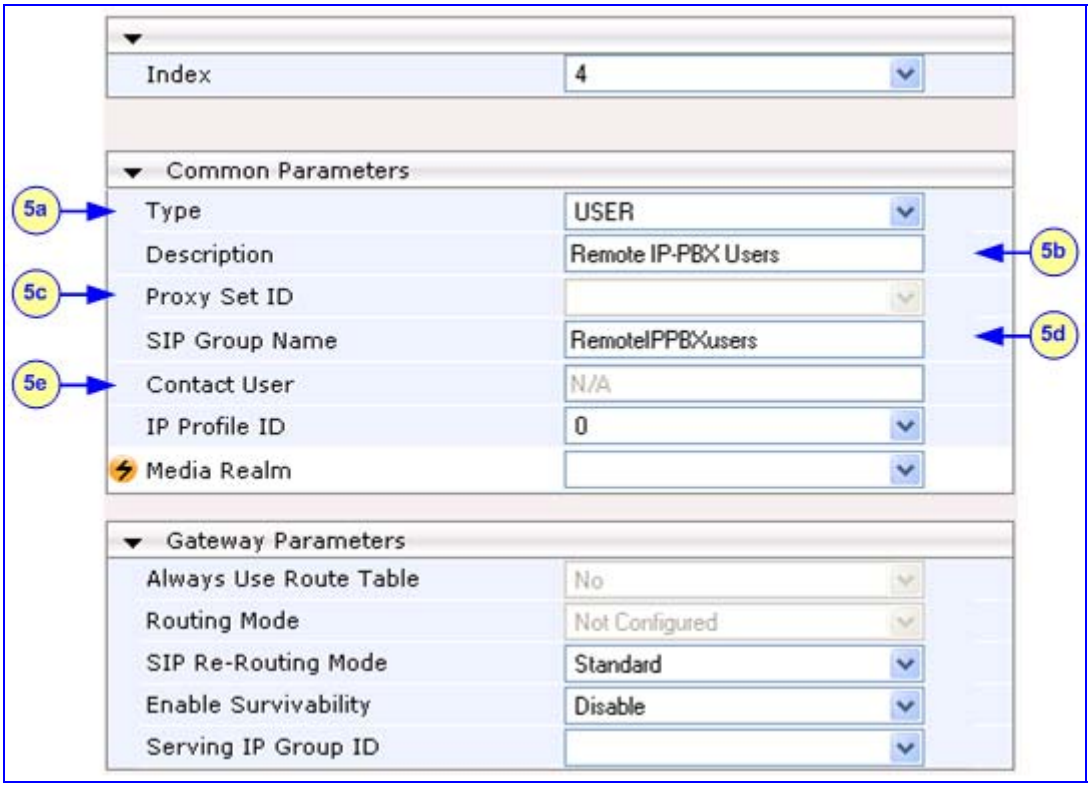
Figure 9-15: Defining IP Group 3



5. Define IP Group **#4** for the remote IP-PBX users:
   a. From the 'Type' drop-down list, select 'USER'.
   b. In the 'Description' field, type an arbitrary name for the IP Group (e.g., IP-PBX).
   c. In the 'SIP Group Name' field, enter the host name that is used internal in the device's database for this IP Group (e.g., RemoteIPPBXusers).

d. From the 'Serving IP Group ID' drop-down list, select "3" (i.e. the IP Group for the IP-PBX).

Figure 9-16: Defining IP Group 4



Note: No Serving IP Groups are defined for ITSP-A and ITSP-B. Instead, the 'Outbound IP Routing' table (refer to "Step 9: Configure Outbound IP Routing" on page 513) is used to configure outbound call routing for calls originating from these ITSP IP Groups.

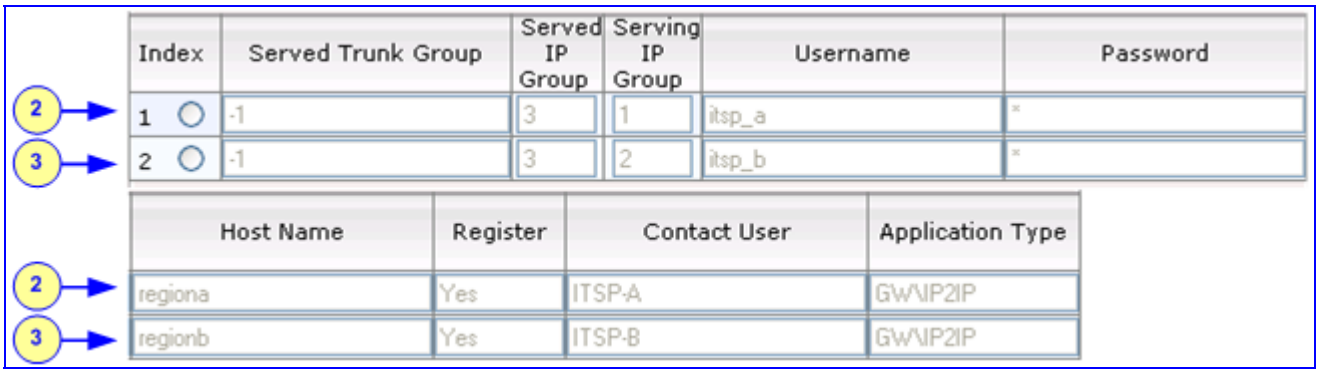
9.2.1.2.6 Step 6: Configure the Account Table

The Account table is used by the device to register to an ITSP on behalf of the IP-PBX. As described previously, the ITSP's requires registration and authentication to provide service. For the example, the Served IP Group is the IP-PBX (IP Group ID #3) and the Serving IP Groups are the two ITSP's (IP Group ID's #1 and #2).

➢ **To configure the Account table:**

1. Open the 'Account Table' page (**Protocol Configuration** menu > **Proxies, Registration, IP Groups** submenu > **Account Table**).

Figure 9-17: Defining Accounts for Registration



2. Configure Account ID **#1** for IP-PBX authentication and registration with ITSP-A:
   - In the 'Served IP Group' field, enter '3' to indicate that authentication is performed on behalf of IP Group #3 (i.e., the IP-PBX).
   - In the 'Serving IP Group' field, enter '1' to indicate that registration/authentication is with IP Group #1 (i.e., ITSP-A).
   - In the 'Username', enter the SIP username for authentication supplied by ITSP-A (e.g., itsp_a).
   - In the 'Password' field, enter the SIP password for authentication supplied by ITSP-A (e.g., 12345).
   - In the 'Register' field, enter '1' to enable registration with ITSP-A.
3. Configure Account ID **#2** for IP-PBX registration) with ITSP-A Registrar server:
   - In the 'Served IP Group' field, enter '3' to indicate that registration is performed on behalf of IP Group #3 (i.e., the IP-PBX).
   - In the 'Serving IP Group' field, enter '2' to indicate that registration is with IP Group #3 (e.g., ITSP-B).
   - In the 'Username', enter the SIP username for the registration/authentication supplied by ITSP-B (e.g., itsp_b).
   - In the 'Password' field, enter the SIP password for registration/authentication supplied by ITSP-B (e.g., 11111).
   - In the 'Register' field, enter '1' to enable registration with ITSP-B.

9.2.1.2.7 Step 7: Configure IP Profiles for Voice Coders

Since different voice coders are used by the IP-PBX (G.711) and the ITSP's (G.723), you need to define two IP Profiles:

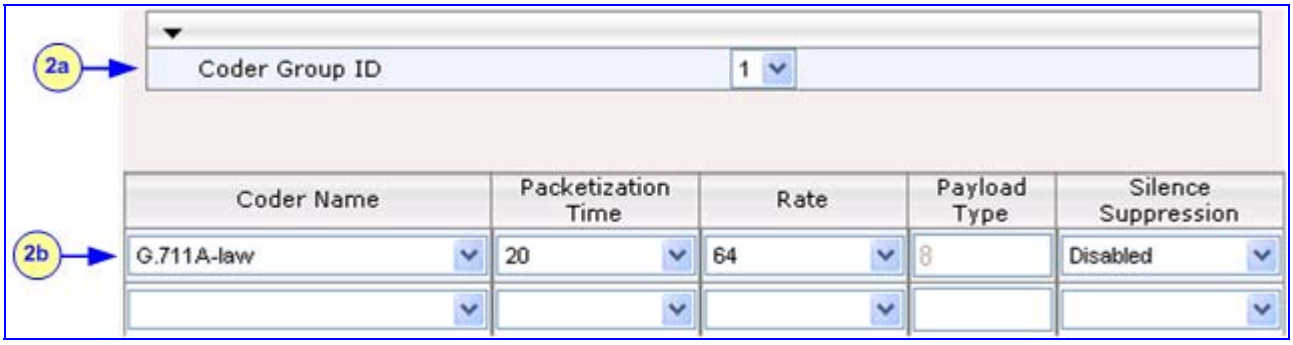
- Profile ID #1 - configured with G.711 for the IP-PBX
- Profile ID #2 - configured with G.723 for the ITSP's

These profiles are later used in the 'Inbound IP Routing' and 'Outbound IP Routing' tables.

➢ **To configure IP Profiles for voice coders:**

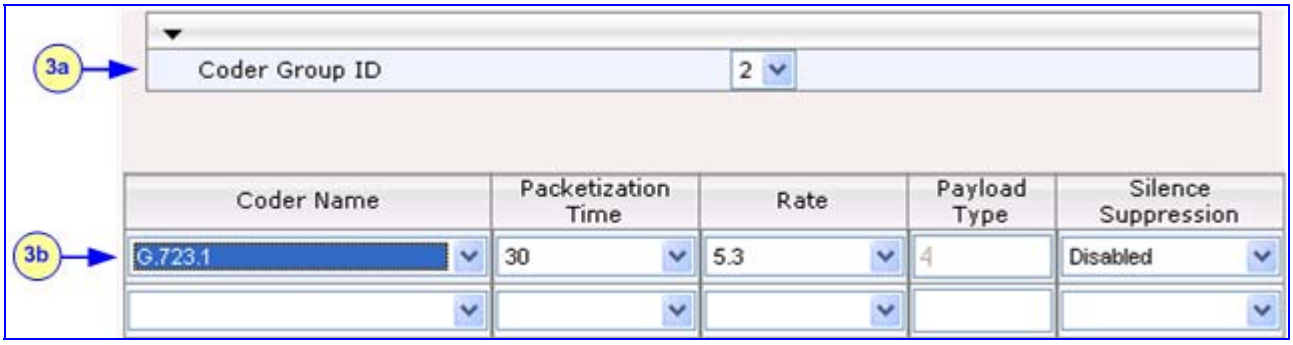
1. Open the 'Coder Group Settings' page (**Protocol Configuration** menu > **Coders And Profile Definitions** submenu > **Coder Group Settings**).
2. Configure Coder Group ID **#1** for the IP-PBX (as shown in the figure below):
   a. From the 'Coder Group ID' drop-down list, select '1'.
   b. From the 'Coder Name' drop-down list, select 'G.711A-law'.
   c. Click **Submit**.

Figure 9-18: Defining Coder Group ID 1



3. Configure Coder Group ID #2 for the ITSP's (as shown in the figure below):
   a. From the 'Coder Group ID' drop-down list, select '2'.
   b. From the 'Coder Name' drop-down list, select 'G.723.1'.
   c. Click **Submit**.

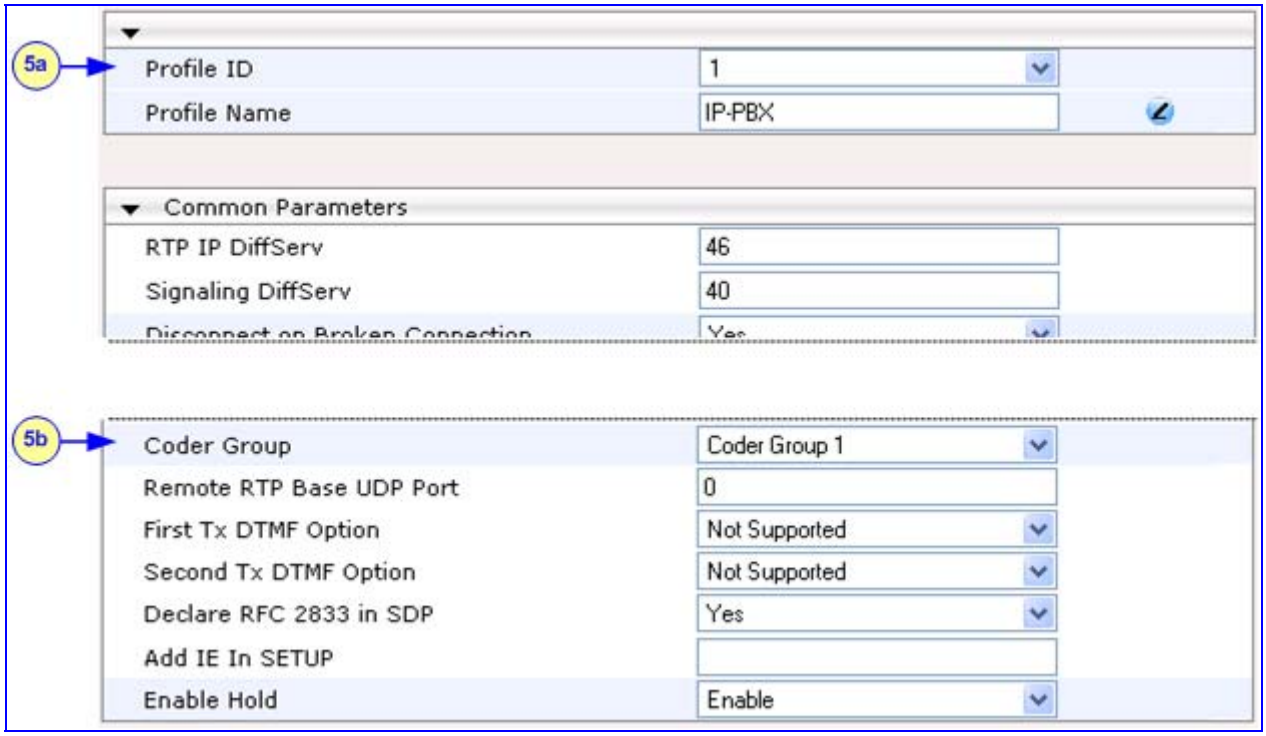
Figure 9-19: Defining Coder Group ID 2



4. Open the 'IP Profile Settings' page (Protocol Configuration menu > Profile Definitions submenu > IP Profile Settings option).

5. Configure Profile ID #1 for the IP-PBX (as shown below):
   a. From the 'Profile ID' drop-down list, select '1'.
   b. From the 'Coder Group' drop-down list, select 'Coder Group 1'.
   c. Click **Submit**.

Figure 9-20: Defining IP Profile ID 1



6. Configure Profile ID #2 for the ITSP's:
   a. From the 'Profile ID' drop-down list, select '2'.
   b. From the 'Coder Group' drop-down list, select 'Coder Group 2'.
   c. Click **Submit**.

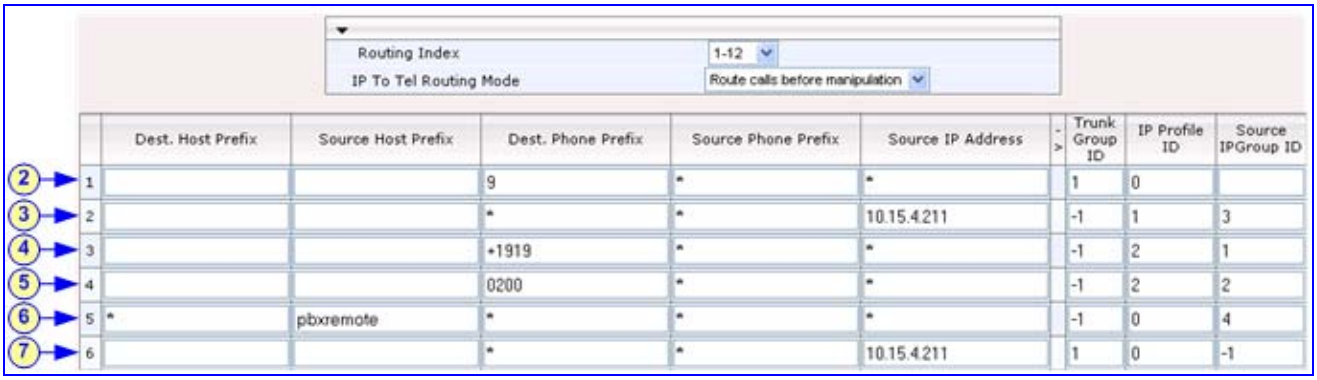
9.2.1.2.8 Step 8: Configure Inbound IP Routing

This step defines how to configure the device for routing inbound (i.e., received) IP-to-IP calls. The table in which this is configured uses the IP Groups that you defined in "Step 5: Configure the IP Groups" on page 504.

➢ **To configure inbound IP routing:**

1. Open the 'Inbound IP Routing Table' page (**Protocol Configuration** menu > **Routing Tables** submenu > **IP to Trunk Group Routing**).

Figure 9-21: Defining Inbound IP Routing Rules


Routing Index
1-12
IP To Tel Routing Mode
Route calls before manipulation
Dest. Host Prefix
Source Host Prefix
Dest. Phone Prefix
Source Phone Prefix
Source IP Address
Trunk Group ID
IP Profile ID
Source IPGroup ID
1 9 * * 1 0
2 * * 10.15.4.211 -1 1 3
3 +1919 * * -1 2 1
4 0200 * * -1 2 2
5 * pbxremote * * * -1 0 4
6 * * 10.15.4.211 1 0 -1

2. **Index #1:** routes calls with prefix 9 (i.e., local calls) dialed from IP-PBX users to the local PSTN:
   - 'Dest Phone Prefix': enter "9" for the dialing prefix for local calls.
   - 'Trunk Group ID': enter "1" to indicate that these calls are routed to the Trunk (belonging to Trunk Group #1) connected between the device and the local PSTN network.
3. **Index #2:** identifies IP calls received from the IP-PBX as IP-to-IP calls and assigns them to the IP Group ID configured for the IP-PBX:
   - 'Dest Phone Prefix': enter the asterisk (*) symbol to indicate all destinations.
   - 'Source IP Address': enter the IP address of the IP-PBX (i.e., 10.15.4.211).
   - 'Trunk Group ID': enter "-1" to indicate that these calls are IP-to-IP calls.
   - 'IP Profile ID': enter "1" to assign these calls to Profile ID #1 to use G.711.
   - 'Source IP Group ID': enter "3" to assign these calls to the IP Group pertaining to the IP-PBX.
4. **Index #3:** identifies IP calls received from ITSP-A as IP-to-IP calls and assigns them to the IP Group ID configured for ITSP-A:
   - 'Dest Phone Prefix': ITSP-A assigns the Enterprise a range of numbers that start with +1919. Enter this prefix to indicate calls received from this ITSP.
   - 'Trunk Group ID': enter "-1" to indicate that these calls are IP-to-IP calls.
   - 'IP Profile ID': enter "2" to assign these calls to Profile ID #2 to use G.723.
   - 'Source IP Group ID': enter "1" to assign these calls to IP Group pertaining to ITSP-A.

5. **Index #4:** identifies IP calls received from ITSP-B as IP-to-IP calls and assigns them to the IP Group ID configured for ITSP-B:
   - 'Dest Phone Prefix': ITSP-B assigns the Enterprise a range of numbers that start with 0200. Enter this prefix to indicate calls coming from this ITSP.
   - 'Trunk Group ID': enter "-1" to indicate that these calls are IP-to-IP calls.
   - 'IP Profile ID': enter "2" to assign these calls to Profile ID #2 to use G.723.
   - 'Source IP Group ID': enter "2" to assign these calls to IP Group pertaining to ITSP-B.
6. **Index #5:** identifies all IP calls received from IP-PBX remote users:
   - 'Source Host Prefix': enter "PBXuser". This is the host name that appears in the From header of the Request URI received from remote IP-PBX users.
   - 'Trunk Group ID': enter "-1" to indicate that these calls are IP-to-IP calls.
   - 'Source IP Group ID': enter "4" to assign these calls to the IP Group pertaining to the remote IP-PBX users.
7. **Index #6:** is used for alternative routing. This configuration identifies all IP calls received from the IP-PBX and which can't reach the ITSP's servers (e.g. loss of connection with ITSP's) and routes them to the local PSTN network:
   - 'Dest Phone Prefix': enter the asterisk (*) symbol to indicate all destinations.
   - 'Source IP Address': enter the IP address of the IP-PBX (i.e., 10.15.4.211).
   - 'Trunk Group ID': enter "1" to route these calls to the Trunk Group ID configured for the Trunk connected to the device and interfacing with the local PSTN.
   - 'Source IP Group ID': enter "-1" to indicate that these calls are not assigned to any source IP Group.

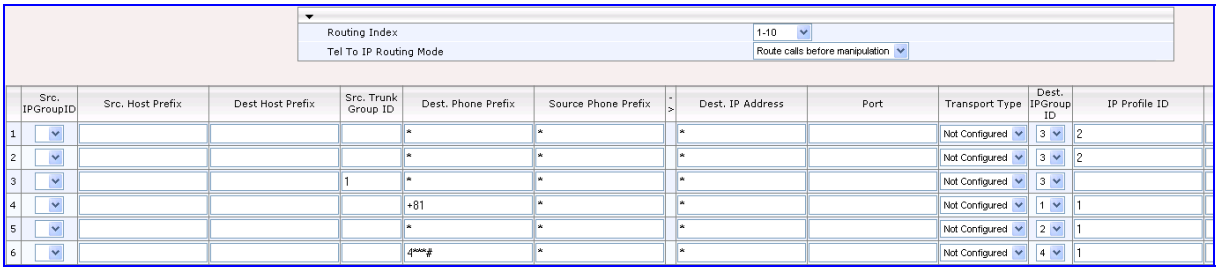
9.2.1.2.9 Step 9: Configure Outbound IP Routing

This step defines how to configure the device for routing outbound (i.e., sent) IP-to-IP calls. In our example scenario, calls from both ITSP's must be routed to the IP-PBX, while outgoing calls from IP-PBX users must be routed according to destination. If the calls are destined to the Japanese market, then they are routed to ITSP-B; for all other destinations, the calls are routed to ITSP-A. This configuration uses the IP Groups defined in "Step 5: Configure the IP Groups" on page 504 and IP Profiles defined in "Step 7: Configure IP Profiles for Voice Coders" on page 509.

➢ **To configure outbound IP routing rules:**

1. Open the 'Outbound IP Routing Table' page (**Protocol Configuration** menu > **Routing Tables** submenu > **Tel to IP Routing**).

Figure 9-22: Defining Outbound IP Routing Rules


Routing Index
1-10
Tel To IP Routing Mode
Route calls before manipulation
Src. IPGroupID
Src. Host Prefix
Dest Host Prefix
Src. Trunk Group ID
Dest. Phone Prefix
Source Phone Prefix
Dest. IP Address
Port
Transport Type
Dest. IPGroup ID
IP Profile ID
1 * * * Not Configured 3 2
2 * * * Not Configured 3 2
3 1 * * * Not Configured 3
4 +81 * * Not Configured 1 1
5 * * * Not Configured 2 1
6 4***# * * Not Configured 4 1

2. **Index #1:** routes IP calls received from ITSP-A to the IP-PBX:
   - 'Source IP Group ID': select "1" to indicate received (inbound) calls identified as belonging to the IP Group configured for ITSP-A.
   - 'Dest Phone Prefix' and 'Source Phone Prefix' : enter the asterisk (*) symbol to indicate all destinations and callers respectively.
   - 'Dest IP Group ID': select "3" to indicate the destination IP Group to where these calls are sent, i.e., to the IP-PBX.
   - 'IP Profile ID': enter "2" to indicate the IP Profile configured for G.723.
3. **Index #2:** routes IP calls received from ITSP-B to the IP-PBX:
   - 'Source IP Group ID': select "2" to indicate received (inbound) calls identified as belonging to the IP Group configured for ITSP-B.
   - 'Dest Phone Prefix' and 'Source Phone Prefix': enter the asterisk (*) symbol to indicate all destinations and callers respectively.
   - 'Dest IP Group ID': select "3" to indicate the destination IP Group to where these calls are sent, i.e., to the IP-PBX.
   - 'IP Profile ID': enter "2" to indicate the IP Profile configured for G.723.
4. **Index #3:** routes calls received from the local PSTN network to the IP-PBX:
   - 'Source Trunk Group ID': enter '1' to indicate calls received on the trunk connecting the device to the local PSTN network.
   - 'Dest IP Group ID': select "3" to indicate the destination IP Group to where the calls must be sent, i.e., to the IP-PBX.
5. **Index #4:** routes IP calls received from the IP-PBX to ITSP-A:
   - 'Source IP Group ID': select "3" to indicate received (inbound) calls identified as belonging to the IP Group configured for the IP-PBX.
   - 'Dest Phone Prefix': enter the +81 to indicate calls to Japan (i.e., with prefix +81).
   - 'Source Phone Prefix': enter the asterisk (*) symbol to indicate all sources.
   - 'Dest IP Group ID': select "1" to indicate the destination IP Group to where the calls must be sent, i.e., to ITSP-A.
   - 'IP Profile ID': enter "1" to indicate the IP Profile configured for G.711.
6. **Index #5:** routes IP calls received from the IP-PBX to ITSP-B:
   - 'Source IP Group ID': select "3" to indicate received (inbound) calls identified as belonging to the IP Group configured for the IP-PBX.
   - 'Dest Phone Prefix' and 'Source Phone Prefix': enter the asterisk (*) symbol to indicate all destinations (besides Japan) and all sources respectively.
   - 'Dest IP Group ID': select "2" to indicate the destination IP Group to where the calls must be sent, i.e., to ITSP-A.
   - 'IP Profile ID': enter "1" to indicate the IP Profile configured for G.711.

7. **Index #6:** routes dialed calls (four digits starting with digit 4) from IP-PBX to remote IP-PBX users. The device searches its database for the remote users registered number, and then sends an INVITE to the remote user's IP address (listed in the database):
   - 'Source IP Group ID': select "3" to indicate received (inbound) calls identified as belonging to the IP Group configured for the IP-PBX.
   - 'Dest Phone Prefix': enter the digit "4xxx#" to indicate all calls dialed from IP-PBX that include four digits and start with the digit 4.
   - 'Dest IP Group ID': select "4" to indicate the destination IP Group to where the calls must be sent, i.e., to remote IP-PBX users.
   - 'IP Profile ID': enter "1" to indicate the IP Profile configured for G.711.

9.2.1.2.10 Step 10: Configure Destination Phone Number Manipulation

This step defines how to manipulate the destination phone number. The IP-PBX users in our example scenario use a 4-digit extension number. The incoming calls from the ITSP's have different prefixes and different lengths. This manipulation leaves only the four digits of the user's destination number coming from the ITSP's.

➢ **To configure destination phone number manipulation:**

1. Open the 'Destination Phone Number Manipulation Table for IP -> Tel calls' page (**Protocol Configuration** menu > **Manipulation Tables** submenu > **Dest Number Tel->IP**).

Figure 9-23: Defining Destination Phone Number Manipulation Rules

	Index	Source Trunk Group	Source IP Group	Destination Prefix	Source Prefix	Stripped Digits From Left	Stripped Digits From Right
(2)	1	-1	-1	+1919	*	0	0
(3)	2	-1	-1	0200	*	0	0

	Prefix to Add	Suffix to Add	Number of Digits to Leave
(2)			4
(3)			4

2. **Index #1:** defines destination number manipulation of IP calls received from ITSP-A. The phone number of calls received with prefix +1919 (i.e., from ITSP-A) are removed except for the last four digits:
   - 'Destination Prefix': enter the prefix +1919.
   - 'Source Prefix': enter the asterisk (*) symbol to indicate all sources.
   - 'Number of Digits to Leave': enter "4" to leave only the last four digits.
3. **Index #2:** defines destination number manipulation of IP calls received from ITSP-B. The phone number of calls received with prefix 0200 (i.e., from ITSP-B) are removed except for the last four digits**:**
   - 'Destination Prefix': enter the prefix 0200.
   - 'Source Prefix': enter the asterisk (*) symbol to indicate all sources.
   - 'Number of Digits to Leave': enter "4" to leave only the last four digits.

9.2.2 Stand-Alone Survivability (SAS) Feature

The device's Stand-Alone Survivability (SAS) feature ensures telephony communication continuity (survivability) for enterprises using hosted IP services (such as IP Centrex) or IP-PBX in cases of failure of these entities. In case of failure of the IP Centrex, IP-PBX servers (or even WAN connection and access Internet modem), the enterprise typically loses its internal telephony service at any branch, between its offices, and with the external environment. In addition, typically these failures lead to the inability to make emergency calls (e.g., 911 in North America). Despite these possible point of failures, the device's SAS feature ensures that the enterprise's telephony services (e.g., SIP IP phones or soft phones) are maintained, by routing calls to the PSTN (i.e., providing PSTN fallback).

The maximum number of SAS registered users supported by the device is 200.

The SAS feature operates in one of two modes:

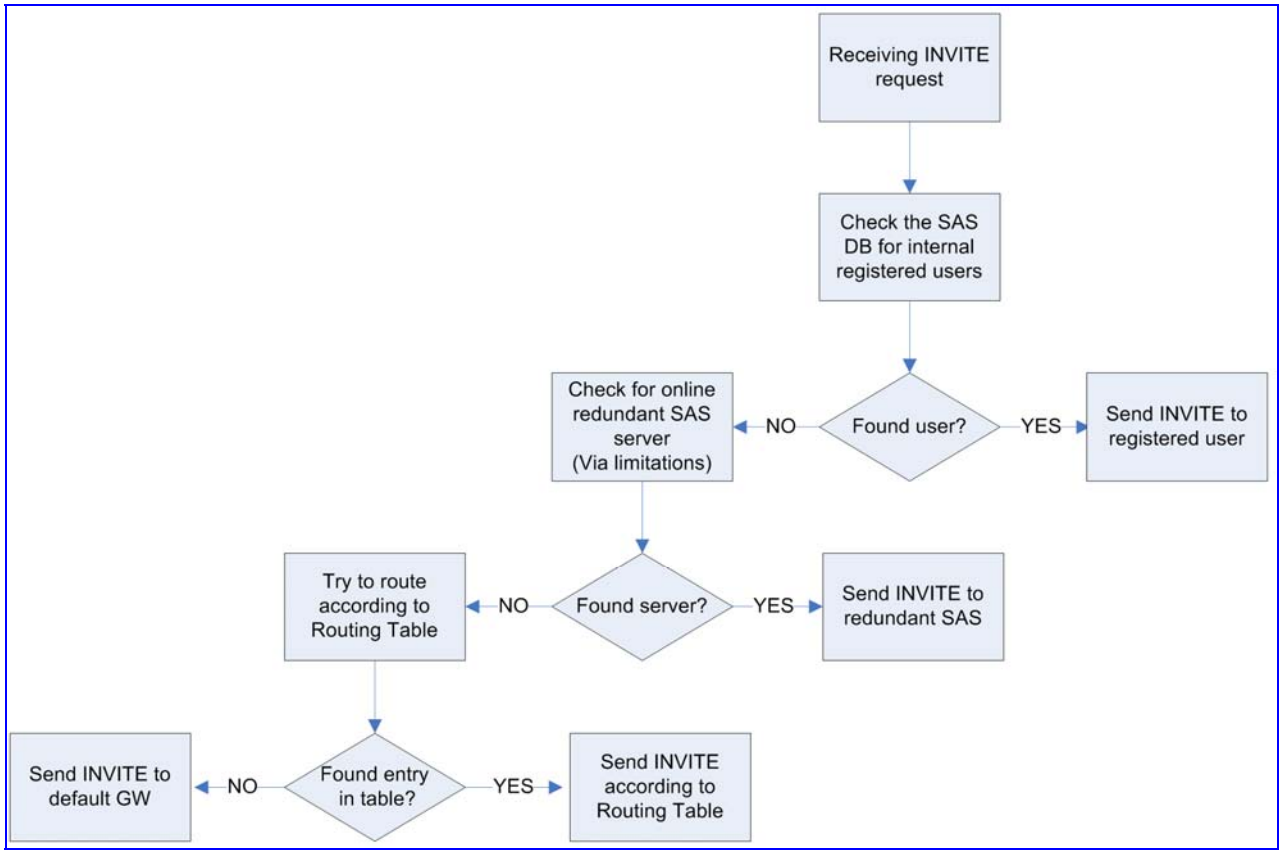
- **Normal:** Initially, the device's SAS agent serves as a registrar (and an outbound Proxy server) to which every VoIP CPE (e.g., IP phones) within the enterprise's LAN registers. The SAS agent at the same time sends all these registration requests to the Proxy server (e.g., IP-Centrex or IP-PBX). This ensures registration redundancy by the SAS agent for all telephony equipment. Therefore, the SAS agent functions as a stateful proxy, passing all SIP requests received from the enterprise to the Proxy and vice versa. In parallel, the SAS agent continuously maintains a keep-alive "handshake" with the Proxy server, using SIP OPTIONS or re-INVITE messages.
- **Emergency:** The SAS agent switches to this mode if it detects (from the keep-alive responses) that the connection with the Proxy is lost. This can occur due to Proxy server failure or WAN problems. In this mode, when the connection with the Proxy server is down, the SAS agent handles all internal calls within the enterprise. In the case of outgoing calls, the SAS agent forwards these to a local VoIP gateway (this can be the device itself or a separate analog or digital gateway). For PSTN fallback, the local VoIP gateway should be equipped with analog (FXO) lines or digital (E1/T1) trunk(s) for PSTN connectivity. In this way, the enterprise preserves its capability for internal and outgoing calls.

The call routing rules for SAS is configured in the 'IP2IP Routing Table' page (refer to "Configuring the IP2IP Routing Table (SAS)" on page 181). This table provides enhanced call routing capabilities (such as built-in ENUM queries and redundant SAS proxy server load balancing) for routing received SIP INVITE messages. When SAS receives a SIP INVITE request from a Proxy server, the following routing logic is performed:

a. Sends the request according to rules configured in the IP2IP Routing table.
b. If no matching routing rule exists, the device sends the request according to its SAS registration database.
c. If no routing rule is located in the database, the device sends the request according to the Request-URI header.

The received INVITE message is routed as depicted in the flow chart below:

Figure 9-24: SAS Routing in Emergency Mode



9.2.2.1 Configuring SAS

For configuring the device to operate with SAS, perform the following configurations:

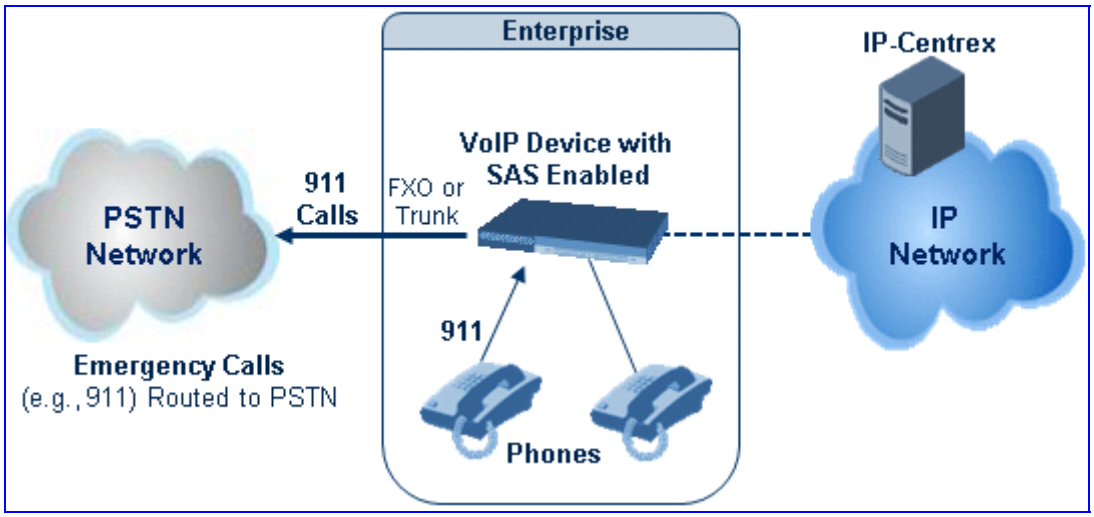
- IsProxyUsed = 1
- ProxyIP 0 = <SAS agent's IP address, i.e., the device>
- ProxyIP 1 = <external Proxy server IP address>
- IsRegisterNeeded = 1 (for the device)
- RegistrarIP = ‘ ‘
- SIPDestinationPort = 5080
- IsUserPhone = 0 (don’t use “user=phone” in SIP URL)
- IsUserPhoneInFrom = 0 (don’t use “user=phone” in From Header)
- IsFallbackUsed = 0
- EnableProxyKeepAlive = 1 (enables keep-alive with Proxy using OPTIONS)
- EnableSAS = 1
- SASLocalSIPUDPPort = (default 5080)
- SASRegistrationTime = <expiration time that SAS returns in the 200 OK to REGISTER in Emergency mode> (default 20)

- SASDefaultGatewayIP = < SAS gateway IP address>
- SASProxySet = 1
- IP2IPRouting (SAS call routing rules)

9.2.2.2 Configuring SAS Emergency Calls

The device's SAS agent can be configured to detect a user-defined emergency number (e.g. 911 in North America), which it then redirects the call directly to the PSTN (through its FXO interface or E1/T1 trunk). The emergency number is configured using the *ini* file parameter SASEmergencyNumbers (for a detailed description, refer to "SIP Configuration Parameters" on page 284).

Figure 9-25: SAS Routing for Emergency Calls



To configure support for emergency calls, configure the parameters below. The device and the SAS feature are configured independently. If the device and the SAS agent use different proxies, then the device's proxy server is defined using the 'Use Default Proxy' parameter, while the SAS proxy agent is defined using the 'Proxy Set' table and SASProxySet parameter.

- EnableSAS = 1
- SASLocalSIPUDPPort = (default 5080)
- IsProxyUsed = 1
- ProxyIP 0 = <external proxy IP address (device)>
- ProxyIP 1 = <external proxy IP address (SAS)>
- IsRegisterNeeded = 1 (for the device)
- IsFallbackUsed = 0
- SASRegistrationTime = <expiration time that SAS returns in the 200 OK to REGISTER in Emergency mode> (default 20)
- SASDefaultGatewayIP = < SAS gateway IP address>
- SASProxySet = 1

9.2.3 DSP Channel Resources for SBC/IP-to-IP/IP Media Functionality

The device supports the IP-to-IP call routing application as well as IP media capabilities. The device provides the required DSP resources (channels) for these applications (in addition to the DSP resources needed for the PRI Trunk interfaces). The device provides flexibility in making DSP resources readily available for these applications, employing a method whereby DSP resources are obtained from the interface module itself (Media Processing Module - MPM), as well as DSP resources "borrowed" from the TRUNKS modules (PRI interfaces).

Notes:

- For the IP-to-IP call routing application, each IP-to-IP call session includes two legs, utilizing two DSP resources.
- In some scenarios, IP-to-IP routing also has the capability of not requiring DSP's.

To enable the IP-to-IP call routing, and/or IP media applications, and allow optimal management of the required DSP resources, the following needs to be addressed:

- Presence of appropriate Software Upgrade Key
- Suitable hardware configuration
- Correct *ini* file configuration

The issues listed above are described in detail below:

1. **Software Upgrade Keys:**

   Verify (by using the Web interface or downloaded *ini* file) that your device has been supplied with the following Software Upgrade Keys:

   - **Number of IPmedia Channels:** this Software Upgrade Key is configured to the maximum number of required DSP resources (e.g., the *ini* file displays "IPMediaDspCh=60").
   - **VoicePromptAnnounc(H248.9):** applicable only to IP media capabilities (all).
   - **Conf:** applicable only to conferencing IP media capabilities.
   - **SBC=<number>:** defines the number of SIP B2BUA sessions (one session for both legs) and whether the B2BUA sessions are IP-to-IP or SBC (only signaling sessions with no relation to the RTP flow). (Applicable only to the IP-to-IP call routing and SBC applications.)

2. **Hardware Configuration:**

   The device can obtain DSP resources for these applications using one of the following hardware configurations:

   - **Media Processing Module (MPM) Modules:** provide DSP resources for IP-to-IP routing, and/or IP media channels for conferencing and IP media functionality. The device can house up to three MPM modules. The DSP resources allocation is as follows:
     - **Without Conferencing:** when the MPM modules are housed in chassis slots 3, 4, and 5, up to 120 DSP resources (without call conferencing) are supported. Each module provides up to 40 DSP resources.

- **With Conferencing:** when the MPM modules are housed in chassis slots 4, 5, and 6, up to 100 DSP resources are supported with call conferencing (up to 60 conference participants). These channels are allocated as follows:
  - ✓ MPM module in Slot 6 provides 20 channels (and enables conferencing for the device)
  - ✓ MPM modules in slots 4 and 5 each provide 40 channels (i.e., total of 80 channels)

Note: If the device houses all three MPM modules, no other interface module can be housed in the device.

- **PRI Modules:** DSP resources can be obtained from existing TRUNKS modules (i.e., some PSTN interfaces are "disabled").

  Note: DSP resources cannot be "borrowed" from PSTN interfaces using CAS.

- **Combination of MPM and PRI Modules:** DSP resources can be obtained from the MPM modules and from available (not used) DSP channels on the TRUNKS modules (without affecting the PRI spans).

For example, to achieve 120 channels (with conferencing), you need to use two MPM modules (inserted in slots 5 and 6) as well as one TRUNKS module providing two PRI spans (in Slot 1).

3. **Ini File:**

The *ini* file must be configured with the following *ini* file parameters:

- **EnableIP2IPApplication:** set to 1 if you want to enable the IP-to-IP call routing application.
- **EnableIPMediaChannels:** set to 1. This (together with the IPMediaDspCh Software Upgrade Key) reduces the number of DSP channels per TRUNKS module. Each DSP typically provides 24 channels for the PRI interface, but this is reduced to 20 channels as described below:
  - TRUNKS module with one Trunk provides 40 channels (2 DSPs * 20 channels), instead of 48
  - TRUNKS module with two Trunks provides 60 channels (3 DSPs * 20 channels), instead of 72
  - TRUNKS module with four Trunks provides 100 channels (5 DSPs * 20 channels), instead of 120 - cannot be used with MPM modules
- **MediaChannels:** set to the maximum number of required IP media channels (regardless of the module from where the channels are acquired).

Note: Setting the parameter IPmediaChannels to a value that is greater than the available DSP resources (from the MPM and those allocated from the TRUNKS modules), can result in the "stealing" of DSP resources from the B-channels of the PRI spans.

- **IPmediaChannels:** defines the number of DSP channels that are “borrowed” (used) from each TRUNKS module for IP-to-IP routing, and/or IP media, as shown in the example below:

```
[IPMediaChannels]
FORMAT IPMediaChannels Index = IPMediaChannels ModuleID,
IPMediaChannels DSPChannelsReserved;
IPMediaChannels 1 = 1, 15;
IPMediaChannels 2 = 2, 10;
[\IPMediaChannels]
```

Notes:

- The value of IPMediaChannels_DSPChannelsReserved must be in multiples of 5.
- By default, the MPM module is set to the maximum number of IP media channels (i.e., no need to define it in the IPmediaChannels table).
- By default, a TRUNKS module is set to 0 (i.e., no IP media channels).

For obtaining DSP resources from a TRUNKS module without affecting the PRI channels, the following options exist:

- A one-E1 TRUNKS module provides 10 "spare" channels (set IPMediaChannels_DSPChannelsReserved to 10)
- A one-T1 TRUNKS module provides 16 "spare" channels (set IPMediaChannels_DSPChannelsReserved to 15, due to multiples of 5)
- A two-T1 TRUNKS module provides 12 "spare" channels (set IPMediaChannels_DSPChannelsReserved to 10, due to multiples of 5)

9.2.4 Multiple SIP Signaling/Media Interfaces Environment

The device supports multiple logical SIP signaling interfaces and RTP (media) traffic interfaces. This allows you to separate SIP signaling messages and media traffic between different applications (i.e., SAS, Gateway\IP-to-IP), and/or between different networks (e.g., when working with multiple ITSP's).

This feature uses the following configuration tables:

- Media Realms table (refer to "Media Realms" on page 522)
- SRD table (refer to "Signaling Routing Domain (SRD) Entities" on page 522)
- SIP Interface table (refer to "SIP Interfaces" on page 523)

For an example configuration of multiple SIP signaling and media interfaces, refer to "Configuration Example" on page 524.

9.2.4.1 Media Realms

A Media Realm is a range of UDP ports that is associated with a media IP interface/IP address (defined in the Multiple Interface table). Media Realms allow you to divide a media IP interface into several realms, where each realm is specified by a UDP port range. The pool of media interfaces (i.e., Media Realms) are defined in the SIP Media Realm table (the CpMediaRealm parameter). Once created, the Media Realm can be assigned to other entities for routing (e.g., to an IP Group ID in the 'IP Group' table, and to an SRD in the 'SR' table).

9.2.4.2 Signaling Routing Domain (SRD) Entities

A Signaling Routing Domain (SRD) is a set of definitions of IP interfaces, device resources, SIP behaviors and other definitions that together create (from the IP user's perspective) multiple virtual multi-service gateways from one physical device.

SRD provides the following capabilities:

- Multiple, different SIP signaling (SRD associated with a SIP Interface, described later) and RTP media (associated with a Media Realm) interfaces for multiple Layer-3 networks.
- Ability to operate with multiple gateway customers that may reside either in the same or in different Layer-3 networks as the device. This allows separation of signaling traffic between different customers. In such a scenario, the device is configured with multiple SRD's.

Typically, one SRD is defined for each group of SIP User Agents/UA (e.g. proxies, IP phones, application servers, gateways, softswitches) that communicate with each other. This provides these entities with VoIP services that reside on the same Layer-3 network (must be able to communicate without traversing NAT devices and must not have overlapping IP addresses).

Routing from one SRD to another is possible, where each routing destination (IP Group or destination address) must indicate the SRD to which it belongs.

The configuration of an SRD includes assigning it a unique name and assigning it a Media Realm (media port range associated with a Media IP interface, defined in the SIP Media Realm table) as well as associating it with a SIP Signaling interface (described later). Once configured, the SRD can then be assigned to an IP Group (in the IP Group table) and to a Proxy Set (in the Proxy Set table).

9.2.4.3 SIP Interfaces

A SIP Interface represents one SIP signaling entity, which is a combination of UDP, TCP, and TLS ports relating to one specific IP address (network interface, configured in the Multiple Interface table). The SIP Interface is configured with a corresponding SRD. This allows User Agents on the network to communicate with a specific SRD, using the SIP Interface (signaling interface) associated with it.

Each SRD may be associated with up to two SIP Interfaces (one per application type - SAS, Gateway\IP-to-IP). Each SIP Interface must have a unique signaling port (i.e., no two SIP Interfaces can share the same port - no overlapping).

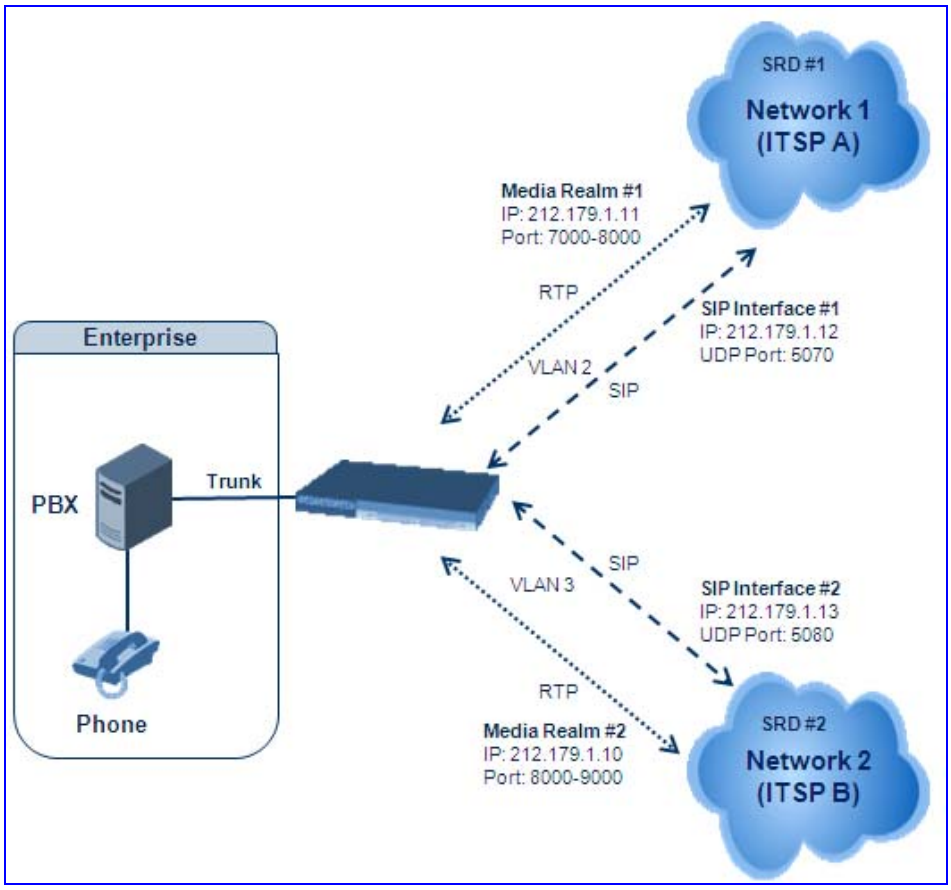
SIP Interfaces are used for the following:

- Defining different SIP signaling ports (listening UDP, TCP, and TLS, and the UDP source ports) for single or multiple interfaces.
- Differentiating between the different application types supported by the device. Only one signaling interface per application type is allowed per SRD.
- Separating signaling traffic of different customers to use different routing tables, manipulations, SIP definitions, etc.

Multiple SIP signaling interfaces are defined in the SIP Interface table (SIPInterface parameter).

The figure below illustrates a typical scenario for implementing multiple SIP signaling interfaces. In this example, different SIP signaling interfaces and RTP traffic interfaces are assigned to Network 1 (ITSP A) and Network 2 (ITSP B).

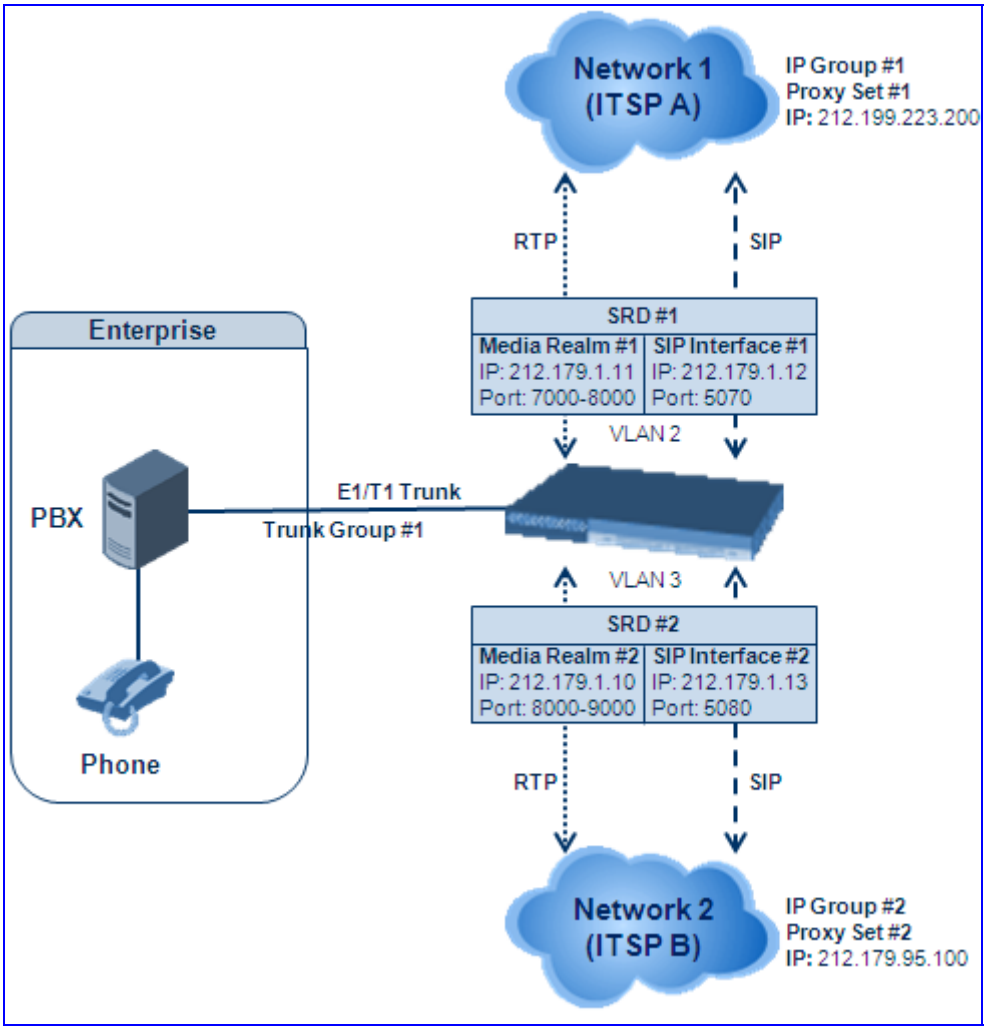
Figure 9-26: Multiple SIP Signaling/RTP Interfaces Scenario



9.2.4.4 Configuration Example

Below is an example configuration for implementing multiple SIP signaling and RTP interfaces. In this example, the device serves as the communication interface between the enterprise's PBX (connected using an E1/T1 trunk) and two ITSP', as shown in the figure below:

Figure 9-27: Multiple SIP Signaling/RTP Interfaces Example

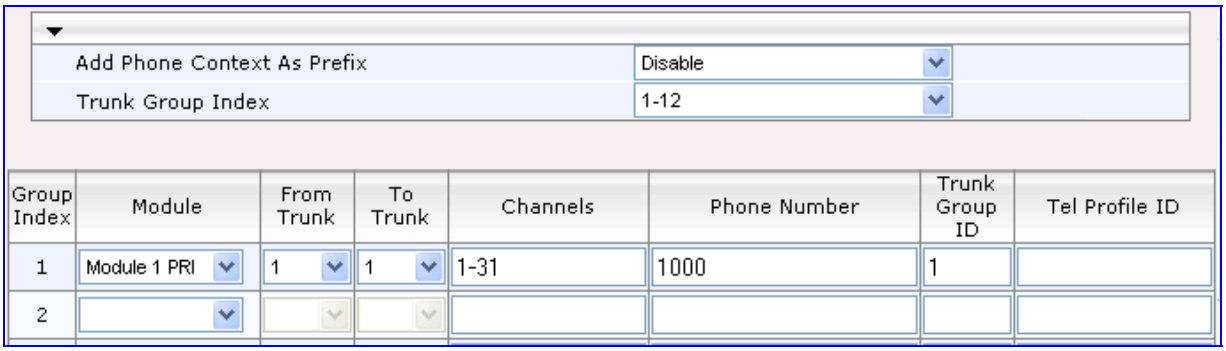


Note that only the steps specific to multi SIP signaling/RTP configuration is shown.

➢ **To configure the scenario example:**

1. Configure Trunk Group ID #1 in the 'Trunk Group Table' page (**Configuration** tab > **Protocol Configuration** menu > **Trunk Group** submenu > **Trunk Group**), as shown in the figure below:

Figure 9-28: Defining Trunk Group



2. Configure the Trunk in the 'Trunk Settings' page (**PSTN Settings** menu > **Trunk Settings**).
3. Configure the IP interfaces in the Multiple Interface table (**Configuration** tab > **Network Settings** menu > **IP Settings**):

Figure 9-29: Defining IP Interfaces

Index	Application Type	IP Address	Prefix Length	Gateway	VLAN ID	Interface Name
0	OAMP + Media + Control	192.168.0.2	24	192.168.0.1	1	Voice
1	Media	212.179.1.11	16	0.0.0.0	2	Media1
2	Media	212.179.1.10	16	0.0.0.0	3	Media2
3	Control	212.179.1.12	16	0.0.0.0	2	SIP1
4	Control	212.179.1.13	16	0.0.0.0	3	SIP2

4. Configure the SIP Media Realms in the 'SIP Media Realm Table' page (**Configuration** tab > **Protocol Configuration** menu > **Media Realm Configuration**):

Figure 9-30: Defining Media Realms

Index	Media Realm Name	IPv4 Interface Name	Port Range Start	Number Of Media Session Legs	Port Range End
1	Realm1	Media1	7000	101	8000
2	Realm2	Media2	8020	20	8210

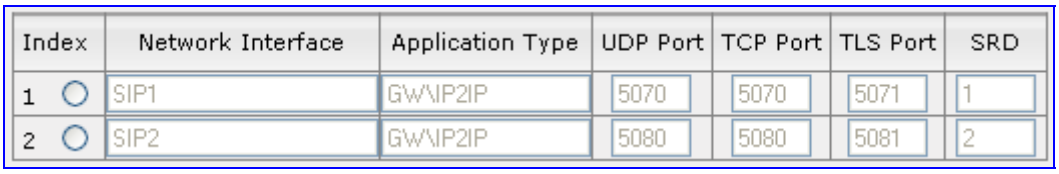
5. Configure the SRD in the SRD table (**Configuration** tab > **Protocol Configuration** menu > **Application Network Settings** submenu > **SRD Table**):

Figure 9-31: Defining SRDs

Index	Name	Media Realm
1	SRD1	Realm1
2	SRD2	Realm2

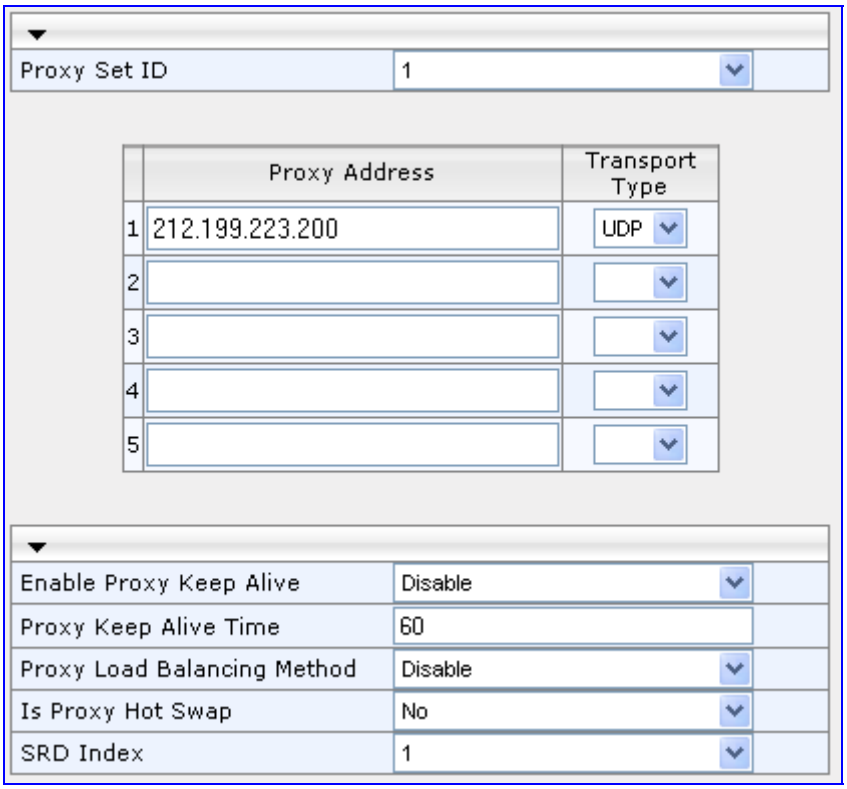
6. Configure the SIP Interfaces in the SIP Interface table (**Configuration** tab > **Protocol Configuration** menu > **Application Network Settings** submenu > **SIP Interface Table**):

Figure 9-32: Defining SIP Interfaces



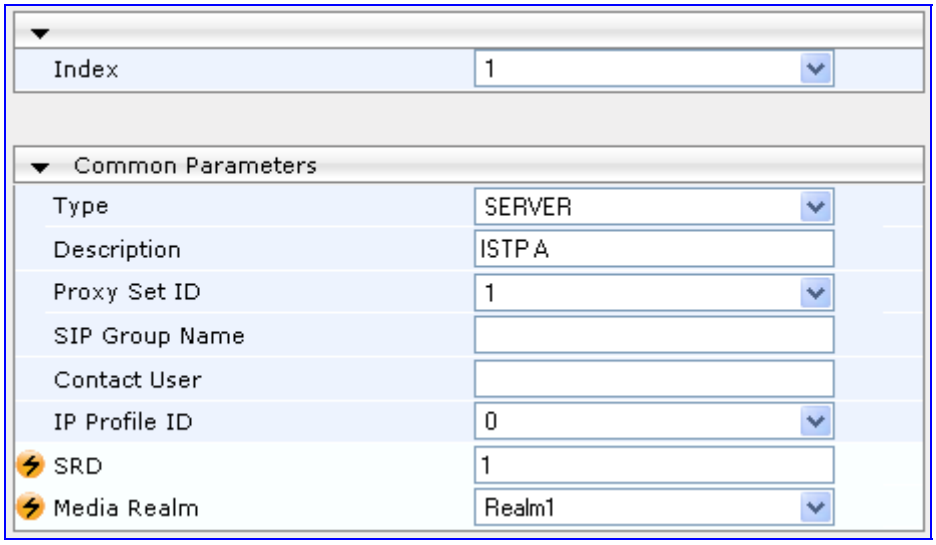
7. Configure Proxy Sets in the Proxy Set table (**Configuration** tab > **Protocol Configuration** menu > **Proxies, Registration, IP Groups** submenu > **Proxy Sets Table**). The figure below configures ITSP A. Do the same for ITSP B, but for Proxy Set 2 with IP address 212.179.95.100 and SRD 2.

Figure 9-33: Defining Proxy Set



8. Configure IP Groups in the IP Groups table (**Configuration** tab > **Protocol Configuration** menu > **Proxies, Registration, IP Groups** submenu > **IP Group Table**). The figure below configures IP group for ITSP A. Do the same for ITSP B, but for Index 2 with SRD 1 and Media Realm to "Realm2".

Figure 9-34: Defining IP Groups



9. Configure IP-to-Trunk Group routing in the Inbound IP Routing table (**Configuration** tab > **Protocol Configuration** menu > **Routing Tables** submenu > **IP to Trunk Group Routing**):

Figure 9-35: Defining IP-to-Trunk Group Routing

	Dest. Host Prefix	Source Host Prefix	Dest. Phone Prefix	Source Phone Prefix	Source IP Address	->	Trunk Group ID
1	*	*	*	*	*		1
2							

10. Configure Trunk Group-to-IP routing (**Configuration** tab > **Protocol Configuration** menu > **Routing Tables** submenu > **Tel to IP Routing** page item):

Figure 9-36: Defining Trunk Group to IP Group Routing

Src. Trunk Group ID	Dest. Phone Prefix	Source Phone Prefix	->	Dest. IP Address	Port	Transport Type	Dest. IPGroup ID
1	[0-1]	*				Not Configured	1
1	*	*				Not Configured	2

9.3 Transcoding using Third-Party Call Control

The device supports transcoding using a third-party call control Application server. This support is provided by the following:

- Using RFC 411C (refer to "Using RFC 4117" on page 528)
- Using RFC 4240 - NetAnn Conferencing (refer to Using RFC 4240 (NetAnn Conferencing) on page 528

Note: Transcoding can also be implemented using the IP-to-IP (IP2IP) application.

9.3.1 Using RFC 4117

The device supports RFC 4117 - Transcoding Services Invocation in the Session Initiation Protocol (SIP) Using Third Party Call Control (3pcc) - providing transcoding services (i.e., acting as a transcoding server). This is used in scenarios where two SIP User Agents (UA) would like to establish a session, but do not have a common coder or media type. When such incompatibilities are found, the UAs need to invoke transcoding services to successfully establish the session. Note that transcoding can also be performed using NetAnn, according to RFC 4240.

To enable the RFC 4117 feature, the parameter EnableRFC4117Transcoding must be set to 1 (and the device must be reset).

The 3pcc call flow is as follows: The device receives from one of the UAs, a single INVITE with an SDP containing two media lines. Each media represents the capabilities of each of the two UAs. The device needs to find a match for both of the media, and opens two channels with two different media ports to the different UAs. The device performs transcoding between the two voice calls.

In the example below, an Application Server sends a special INVITE that consists of two media lines to perform transcoding between G.711 and G.729:

```
m=audio 20000 RTP/AVP 0
c=IN IP4 A.example.com
m=audio 40000 RTP/AVP 18
c=IN IP4 B.example.com
```

9.3.2 Using RFC 4240 (NetAnn Conferencing)

Transcoding is a technology that is used to bridge (translate) between two remote *network* locations, each of which uses a different coder and/or a different DTMF and fax transport types. The device supports IP-to-IP Transcoding. It creates a Transcoding call that is similar to a dial-in, two-party conference call. The SIP URI in the INVITE message is used as a Transcoding service identifier. The Transcoding identifier can be configured using the *ini* file (parameter TranscodingID) or Web interface (refer to "Configuring the IPmedia Parameters" on page 179).

It is assumed that the device is controlled by a third-party, Application server (or any SIP user agent) that instructs the device to start an IP Transcoding call by sending two SIP INVITE messages with SIP URI that includes the Transcoding Identifier name. For example:

```
Invite sip:trans123@audiocodes.com SIP/2.0
```

The left part of the SIP URI includes the Transcoding ID (the default string is 'trans') and is terminated by a unique number (123). The device immediately sends a 200 OK message in response to each INVITE.

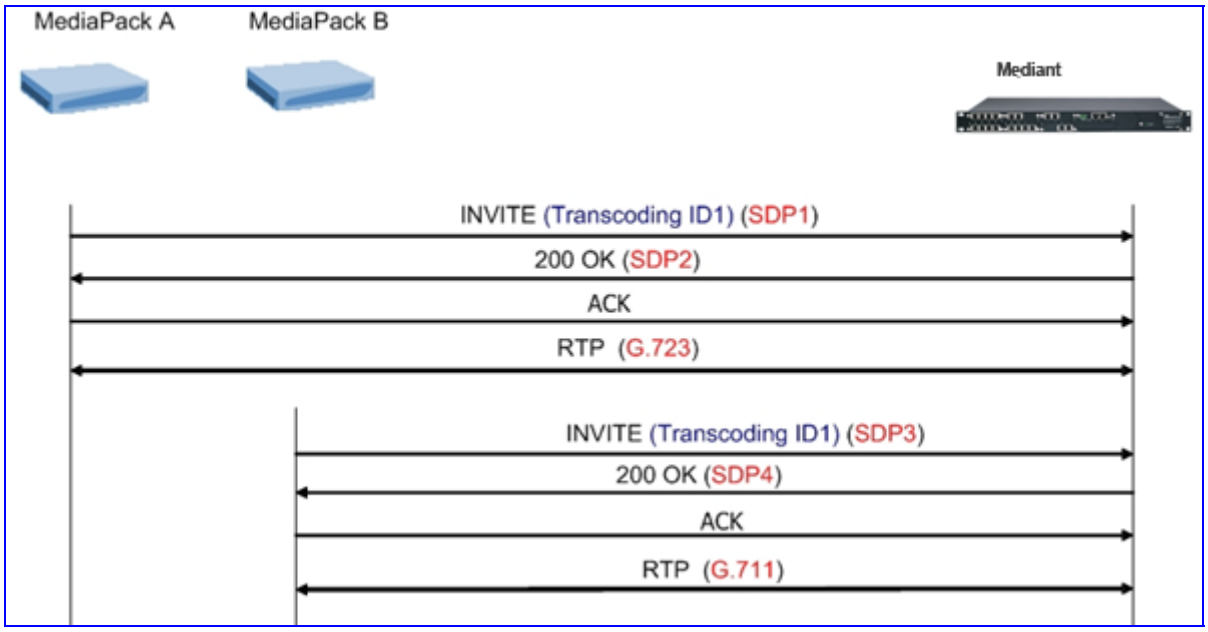
Each of the Transcoding SIP call participants can use a different VoIP coder and a different DTMF transport type, negotiated with the device using common SIP negotiation.

Sending a BYE request to the device by any of the participants, terminates the SIP session and removes it from the Transcoding session. The second BYE from the second participant ends the Transcoding session and releases its resources.

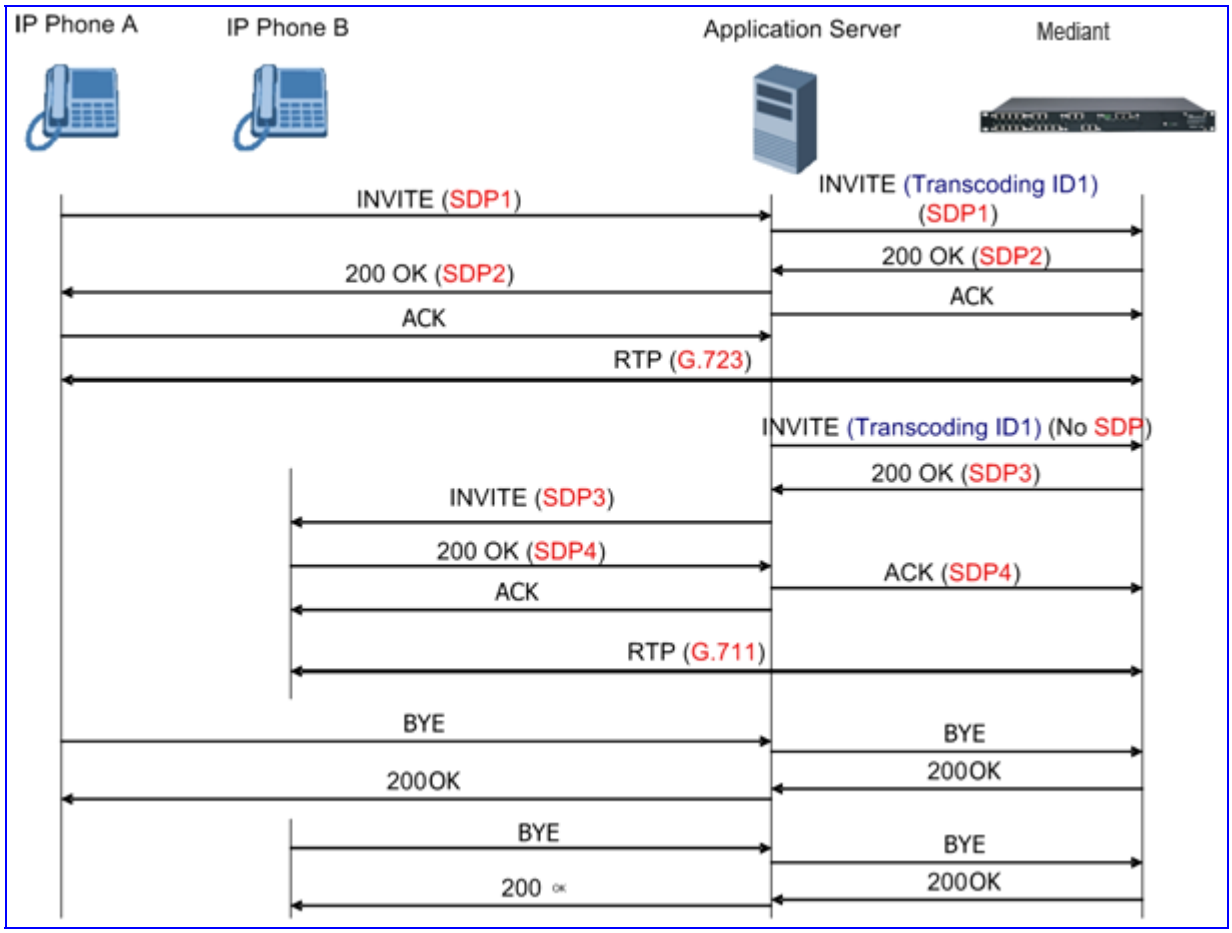
The device uses two media (DSP) channels for each call, thereby reducing the number of available Transcoding sessions to half of the defined value for MediaChannels. To limit the number of resources available for Transcoding, use the *ini* file parameter MediaChannels or Web interface (refer to "Configuring the IPmedia Parameters" on page 179). For example, if MediaChannels is 40, only 20 Transcoding sessions are available.

The figure below illustrates an example of a direct connection to a device:

Figure 9-37: Direct Connection (Example)



The figure below illustrates an example of implementing an Application server:



9.4 Emergency Phone Number Services - E911

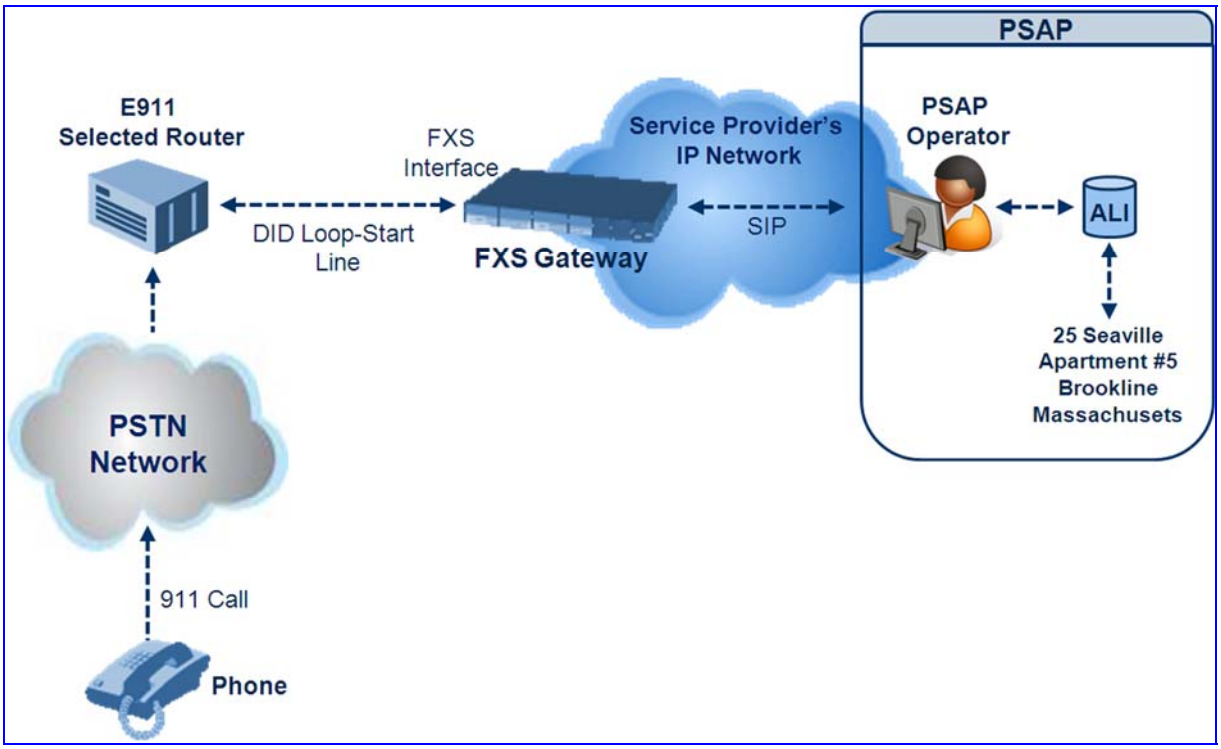
The device supports various emergency phone number services. The device supports the North American emergency telephone number system known as Enhanced 911 (E911), according to the TR-TSY-000350 and Bellcore's GR-350-Jun2003 standards. The E911 emergency system automatically associates a physical address with the calling party's telephone number, and routes the call to the most appropriate (closest) Public Safety Answering Point (PSAP), allowing the PSAP to quickly dispatch emergency response (e.g., police) to the caller's location.

Typically, the dialed emergency number is routed to the appropriate PSAP by the telephone company's switch, known as a 911 Selective Router (or E911 tandem switch). If the PSAP receives calls from the telephone company on old-style digital trunks, they are specially formatted Multi-Frequency (MF) trunks that pass only the calling party's number (known as Automatic Number Identification - ANI). Once the PSAP receives the call, it searches for the physical address that is associated with the calling party's telephone number (in the Automatic Location Identification database - ALI).

9.4.1 FXS Device Emulating PSAP using DID Loop-Start Lines

The FXS device can be configured to emulate PSAP (using DID loop start lines), according to the Telcordia GR-350-CORE specification.

Figure 9-38: FXS Device Emulating PSAP using DID Loop-Start Lines



The call flow of an E911 call to the PSAP is as follows:

1. The E911 tandem switch seizes the line.
2. The FXS device detects the line seize, and then generates a wink signal (nominal 250 msec). The wink can be delayed by configuring the parameter DelayBeforeDIDWink to 200 (for 200 msec or a higher value).
3. The switch detects the wink and then sends the MF Spill digits with ANI and (optional) Pseudo-ANI (P ANI).
4. The FXS device collects the MF digits, and then sends a SIP INVITE message to the PSAP with all collected MF digits in the SIP From header as one string.
5. The FXS device generates a mid-call wink signal (two subsequent polarity reversals) toward the E911 tandem switch upon either detection of an RFC 2833 "hookflash" telephony event, or if a SIP INFO message with a "hooflash" body is received from the PSAP (see the example below). The duration of this "flashhook" wink signal is configured using the parameter FlashHookPeriod (usually 500 msec). Usually the wink signal is followed by DTMF digits sent by PSAP to perform call transfer. Another way to perform the call transfer is to use SIP REFER messages, as described below.
6. The FXS device supports call transfer initiated by the PSAP. If it receives a SIP REFER message with the Refer-To URI host part containing an IP address that is equal to the device's IP address, the FXS device generates a 500-msec wink signal (double polarity reversals), and then (after a user-defined interval configured by the parameter WaitForDialTime), plays DTMF digits according to the transfer number received in the SIP Refer-To header URI userpart.

7. When the call is answered by the PSAP operator, the PSAP sends a SIP 200 OK to the FXS device, and the FXS device then generates a polarity reversal signal to the E911 switch.
8. After the call is disconnected by the PSAP, the PSAP sends a SIP BYE to the FXS device, and the FXS device reverses the polarity of the line toward the tandem switch.

The following parameters need to be configured:

- EnableDIDWink = 1
- EnableReversalPolarity = 1
- PolarityReversalType = 1
- FlashHookPeriod = 500 (for 500 msec "hookflash" mid-call Wink)
- WinkTime = 250 (for 250 msec signalling Wink generated by the FXS device after it detects the line seizure)
- EnableTransfer = 1 (for call transfer)
- LineTransferMode = 1 (for call transfer)
- WaitforDialTime = 1000 (for call transfer)
- SwapTEI2IPCalled&CallingNumbers = 1
- DTMFDetectorEnable = 0
- MFR1DetectorEnable = 1
- DelayBeforeDIDWink = 200 (for 200 msec) - can be configured in the range from 0 (default) to 1000.

Note: Modification of the WinkTime and FlashHookPeriod parameters require a device reset.

The outgoing SIP INVITE message contains the following headers:

```
INVITE sip:Line@DomainName
From: <sip:*81977820#@sipgw>;tag=1c143
To: <sip:Line@DomainName>
```

Where:

- Line = as configured in the Endpoint Phone Number Table.
- SipGtw = configured using the SIPGatewayName parameter.
- From header/user part = calling party number as received from the MF spill.

The ANI and the pseudo-ANI numbers are sent to the PSAP either in the From and/or P-AssertedID SIP header.

Typically, the MF spills are sent from the E911 tandem switch to the PSAP, as shown in the table below:

Table 9-3: Dialed MF Digits Sent to PSAP

Digits of Calling Number	Dialed MF Digits
8 digits "nnnnnnnn" (ANI)	"KPnnnnnnnnST"
12 digits "nnnnnnnnnnnn" (ANI)	"KPnnnnnnnnnnnnSTP"
12 digits ANI and 10 digits PANI	"KPnnnnnnnnnnnnSTKPmmmmmmmmmmST"
two digits "nn"	"KPnnSTP"

The MF KP, ST, and STP digits are mapped as follows:

- * for KP
- # for ST
- B for STP

For example, if ANI and PANI are received, the SIP INVITE contains the following From header:

```
From: <sip:*nnnnnnnnnnnn#*mmmmmmmmmm#@10.2.3.4>;tag=1c14
```

Note: It is possible to remove the * and # characters, using the device's number manipulation rules.

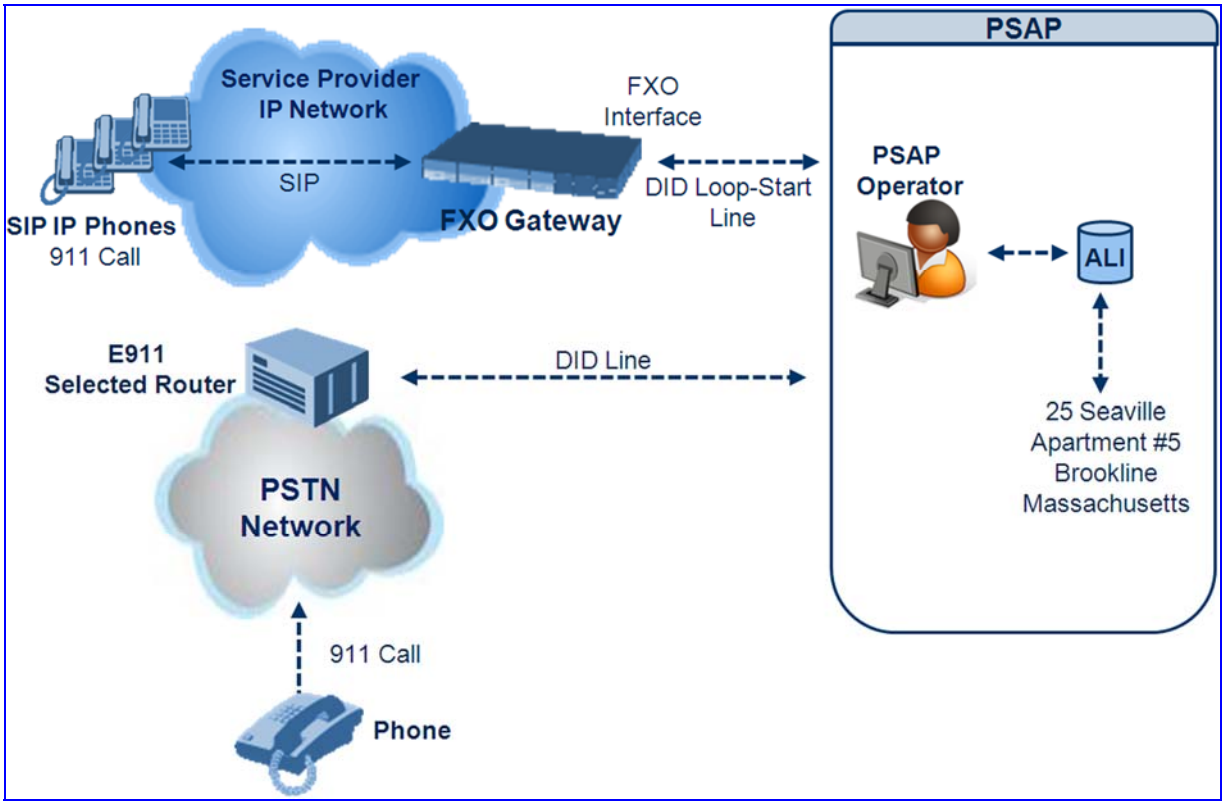
If the device receives the SIP INFO message below, it then generates a "hookflash" mid-call Wink signal:

```
INFO sip:4505656002@192.168.13.40:5060 SIP/2.0
Via:  SIP/2.0/UDP 192.168.13.2:5060
From: port1vega1 <sip:06@192.168.13.2:5060>
To: <sip:4505656002@192.168.13.40:5060>;tag=132878796-
1040067870294
Call-ID: 0010-0016-D69A7DA8-1@192.168.13.2
CSeq:2 INFO
Content-Type: application/broadsoft
Content-Length: 17
event flashhook
```

9.4.2 FXO Device Interworking SIP E911 Calls from Service Provider's IP Network to PSAP DID Lines

The FXO device can interwork SIP emergency E911 calls from the Service Provider's IP network to the analog PSAP DID lines. The standards that define this interface include TR-TSY-000350 or Bellcore's GR-350-Jun2003. This protocol defines signaling between the E911 tandem switch (E911 Selective Router) and the PSAP, using analog loop-start lines. The FXO device can be implemented instead of an E911 switch, by connecting directly to the PSAP DID loop-start lines.

Figure 9-39: FXO Device Interfacing between E911 Switch and PSAP



When an IP phone subscriber dials 911, the device receives the SIP INVITE message and makes a call to the PSAP as follows:

1. The FXO device seizes the line.
2. PSAP sends a Wink signal (250 msec) to the device.
3. Upon receipt of the Wink signal, the device dials MF digits after a user-defined time (WaitForDialTime) containing the caller's ID (ANI) obtained from the SIP headers From or P-Asserted-Identity.
4. When the PSAP operator answers the call, the PSAP sends a polarity reversal to the device, and the device then sends a SIP 200 OK to the IP side.
5. After the PSAP operator disconnects the call, the PSAP reverses the polarity of the line, causing the device to send a SIP BYE to the IP side.
6. If, during active call state, the device receives a Wink signal (typically of 500 msec) from the PSAP, the device generates a SIP INFO message that includes a "hookflash" body, or sends RFC 2833 hookflash Telephony event (according to the HookFlashOption parameter).

7. Following the "hookflash" Wink signal, the PSAP sends DTMF digits. These digits are detected by the device and forwarded to the IP, using RFC 2833 telephony events (or inband, depending on the device's configuration). Typically, this Wink signal followed by the DTMF digits initiates a call transfer.

For supporting the E911 service, used the following configuration parameter settings:

- Enable911PSAP = 1 (also forces the EnableDIDWink and EnableReversalPolarity)
- HookFlashOption = 1 (generates the SIP INFO hookflash message) or 4 for RFC 2833 telephony event
- WinkTime = 700 (defines detection window of 50 to 750 msec for detection of both winks - 250 msec wink sent by the PSAP for starting the device's dialing; 500 msec wink during the call)
- IsTwoStageDial = 0
- EnableHold = 0
- EnableTransfer = 0
  - Use RFC 2833 DTMF relay:
    - RxDTMFOption = 3
    - TxDTMFOption = 4
    - RFC2833PayloadType = 101
- TimeToSampleAnalogLineVoltage = 100
- WaitForDialTime = 1000 (default is 1 sec)

The device expects to receive the ANI number in the From and/or P-Asserted-Identity SIP header. If the pseudo-ANI number exists, it should be sent as the display name in these headers.

Table 9-4: Dialed Number by Device Depending on Calling Number

Digits of Calling Number (ANI)	Digits of Displayed Number	Number Dialed MF Digits
8 "nnnnnnnn"	-	MF dialed "KPnnnnnnnnST"
12 "nnnnnnnnnnnn"	None	"KPnnnnnnnnnnnnSTP"
12 "nnnnnnnnnnnn"	10 "mmmmmmmmmm" (pANI)	"KPnnnnnnnnnnnnSTKPmmmmmmmmmmST"
2 "nn"	None	"KPnnSTP"
1 "n"	-	MF dialed "KPnST" For example: "From: <sip:8>@xyz.com>" generates device MF spill of KP 8 ST

Table Notes:

- For all other cases, a SIP 484 response is sent.
- KP is for *.

- ST is for #.
- STP is for B.

The MF duration of all digits, except for the KP digit is 60 msec. The MF duration of the KP digit is 120 msec. The gap duration is 60 msec between any two MF digits.

Notes:

- Manipulation rules can be configured for the calling (ANI) and called number (but not on the "display" string), for example, to strip 00 from the ANI "00INXXYYYY".
- The called number, received as userpart of the Request URI ("301" in the example below), can be used to route incoming SIP calls to FXO specific ports, using the TrunkGroup and PSTNPrefix parameters.
- When the PSAP party off-hooks and then immediately on-hooks (i.e., the device detects wink), the device releases the call sending SIP response "403 Forbidden" and the release reason 21 (i.e., call rejected) "Reason: Q.850 ;cause=21" is sent. Using the cause mapping parameter, it is possible to change the 403 to any other SIP reason, for example, to 603.
- Sometimes a wink signal sent immediately after the FXO device seizes the line is not detected. To overcome this problem, configure the parameter TimeToSampleAnalogLineVoltage to 100 (instead of 1000 msec, which is the default value). The wink is then detected only after this timeout + 50 msec (minimum 150 msec).

Below are two examples for a) INVITE messages and b) INFO messages generated by hook-flash.

- Example (a): INVITE message with ANI = 333333444444 and pseudo-ANI = 0123456789:

```
  INVITE sip:301@10.33.37.79;user=phone SIP/2.0
Via: SIP/2.0/UDP 10.33.37.78;branch=z9hG4bKac771627168
Max-Forwards: 70
From: "0123456789"
<sip:333333444444@audiocodes.com>;tag=1c771623824
To: <sip:301@10.33.37.79;user=phone>
Call-ID: 771623358412000141530@10.33.37.78
CSeq: 1 INVITE
Contact: <sip:101@10.33.37.78>
Supported: em,100rel,timer,replaces,path
Allow:
REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,INFO,SUB
SCRIBE,UPDATE
User-Agent: Audiocodes-Sip-Gateway-FXO/v.6.00A.020.077
Privacy: none
P-Asserted-Identity: "0123456789" <sip:3333344444@audiocodes.com>
Content-Type: application/sdp
Content-Length: 253

v=0
o=AudiocodesGW 771609035 771608915 IN IP4 10.33.37.78
s=Phone-Call
c=IN IP4 10.33.37.78
t=0 0
m=audio 4000 RTP/AVP 8 0 101
a=rtpmap:8 pcma/8000
a=rtpmap:0 pcmu/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
a=ptime:20
a=sendrecv
```

- Example (b): The detection of a Wink signal generates the following SIP INFO message:

```
INFO sip:4505656002@192.168.13.40:5060 SIP/2.0
Via:  SIP/2.0/UDP 192.168.13.2:5060
From: port1vega1 <sip:06@192.168.13.2:5060>
To: <sip:4505656002@192.168.13.40:5060>;tag=132878796-
1040067870294
Call-ID: 0010-0016-D69A7DA8-1@192.168.13.2
CSeq:2 INFO
Content-Type: application/broadsoft
Content-Length: 17
event flashhook
```

9.5 Routing Based on LDAP Active Directory Queries

The device supports Lightweight Directory Access Protocol (LDAP), allowing the device to make call routing decisions based on information stored on a third-party LDAP server (or Microsoft's Active Directory-based enterprise directory server). This feature enables the usage of one common, popular database to manage and maintain information regarding user's availability, presence, and location.

The LDAP feature can be configured using the *ini* file, Web interface, SNMP, and CLI (for debugging only).

9.5.1 LDAP Overview

The basic LDAP mechanism is described below:

- **Connection:** The device connects and binds to the remote LDAP server either during the service's initialization (at device start-up) or whenever the LDAP server's IP address and port is changed. Service makes 10 attempts to connect and bind to the remote LDAP server with a timeout of 20 seconds between attempts. If connection fails, the service remains in disconnected state until either the LDAP server's IP address or port is changed.

  If connection to the LDAP server later fails, the service attempts to reconnect, as described previously. The SNMP alarm acLDAPLostConnection is sent when connection is broken. Upon successful reconnection, the alarm is cleared.

  Binding to the LDAP server can be anonymous or not. For anonymous binding, the LDAPBindDN and LDAPPassword parameters must not be defined or set to an empty string.

  The address of the LDAP server can be a DNS name (using the LDAPServerName parameter) or an IP address (using the LDAPServerIP parameter).

- **Search:** To run a search using the LDAP service, the path to the directory's subtree where the search is to be performed must be defined (using the LDAPSearchDN parameter). In addition, the search key (known as "filter" in LDAP references), which defines the exact DN to be found and one or more attributes whose values should be returned, must be defined. The device supports up to 20 LDAP search requests.

  If connection to the LDAP server is disrupted during the search, all search requests are dropped and an alarm indicating a failed status is sent to client applications.

- **CLI:** The LDAP CLI is located in the directory IPNetworking\OpenLdap. The following commands can be used:

  - LdapSTatus - displays connection status
  - LdapSearch - searches an LDAP server
  - LDapOpen - opens connection to the LDAP server using parameters provided in configuration file
  - LDapSetDebugmode - sets the LdapDebugLevelMode parameter
  - LDapGetDebugmode – gets the LdapDebugLevelMode parameter value

Relevant parameters: LDAPServiceEnable; LDAPServerIP; LDAPServerDomainName; LDAPServerPort; LDAPPassword; LDAPBindDN; LDAPSearchDN; LDAPDebugMode; LDAPServerMaxRespondTime.

9.5.2 AD-Based Tel-to-IP Routing in Microsoft OCS 2007 Environment

Typically, enterprises wishing to deploy Microsoft's Office Communication Server 2007 (OCS 2007) are faced with a complex, call routing dial plan when migrating users from their existing PBX/IP-PBX to the OCS 2007 platform. As more and more end-users migrate to the new voice system, dialing plan management and PBX link capacity can be adversely impacted. Moreover, it's easy to perceive that even a temporary failure (or disconnection) of Microsoft's Office Communications Server 2007 Mediation Server (Mediation Server) results in no incoming voice calls from the PBX/IP-PBX/PSTN and therefore, it will be impossible to reach the user on the user's Microsoft Office Communicator (OC) client.

This feature enables the device to make Tel-to-IP call routing decisions based on information stored on Microsoft's Active Directory-based (AD) enterprise directory server. This implements one common, central database to manage and maintain information regarding user's availability, presence, and location.

Based on queries sent to the AD, this feature allows you to route incoming Tel calls to one of the following IP domains:

- PBX/IP-PBX (for users yet to migrate to the OCS 2007 platform)
- OCS clients (clients connected to the OCS 2007 platform)
- Mobile

The device queries the AD using the destination number. The device's AD queries return up to three user phone number IP destinations, each pertaining to one of the IP domains listed above. The device routes the call according to the following priority:

1. **OCS SIP address:** The call is routed to Mediation Server (which then routes the call to the OCS client).
2. **Mobile number:** If the Mediation Server or OCS client is unavailable (e.g., SIP response 404 "Not Found" upon INVITE sent to OCS client), the device routes the call to the user's mobile number (if exists in the AD).
3. **PBX/IP-PBX number:** If no OCS client exits in the AD, then the device routes the call to the PBX/IP-PBX (if this fails, the call is routed to the mobile number, if exists).

For enterprises implementing a PBX/IP-PBX system but yet to migrate to the OCS 2007 platform, if the PBX/IP-PBX system is unavailable, the device queries the AD for the users mobile phone number and then routes the call, through the PSTN to the mobile destination.

This feature is configured in the Outbound IP Routing table, where the "LDAP" keywords are entered for the destination phone prefixes. For each IP domain (listed above), the destination numbers are prefixed (case-sensitive) as follows:

- **OCS client number:** "OCS:"
- **PBX number:** "PBX:"
- **Mobile number:** "MOBILE:"
- **LDAP failure:** "LDAP_ERR:"

Note that these prefixes are only involved in the routing and manipulation stages; they are not used as the final destination number.

In addition, once you have configured the LDAP parameters (refer to "LDAP Overview" on page 538), you need to enter the "LDAP" value for the destination IP address of the LDAP server in the Outbound IP Routing table.

For enabling alternative routing, you need to enable the alternative routing mechanism and configure corresponding SIP reasons for alternative routing. For this feature, alternative routing always starts again from the top of the table (first routing rule entry) and not from the next row.

This feature uses the following parameters to configure the attribute names in the AD used in the LDAP query:

- AD attribute for Mediation Server: MSLDAPOCSNumAttributeName (the default is "msRTCSIPPrimaryUserAddress")
- AD attribute for PBX/IP-PBX: MSLDAPPBXNumAttributeName (the default is "telephoneNumber")
- AD attribute for mobile number: MSLDAPMobileNumAttributeName (the default is "mobile")

Below is an example for configuring AD-based routing rules in the Outbound IP Routing Table:

Figure 9-40: Active Directory-based Routing Rules in Outbound IP Routing Table

Src. Trunk Group ID	Dest. Phone Prefix	Source Phone Prefix	->	Dest. IP Address	Port
*	PBX:	*		10.33.45.65	
*	OCS:	*		10.33.45.68	
*	MOBILE:	*		10.33.45.100	
*	LDAP_ERR	*		10.33.45.80	
*	*	*		LDAP	
*	*	*		10.33.45.72	

- **First rule:** sends call to IP-PBX (10.33.45.65) if AD query replies with prefix "PBX:"
- **Second rule:** sends call to OCS client (i.e., Mediation Server at 10.33.45.68) if AD query replies with prefix "OCS:"
- **Third rule:** sends call to users mobile phone number (to PSTN through the device's IP address, 10.33.45.100) if AD query replies with prefix "MOBILE:"
- **Fourth rule:** sends call to IP address of device, for example (10.33.45.80) if no response from LDAP server
- **Fifth rule:** sends query of received Tel destination number to LDAP server, and then routes the call according to query reply and routing rules at top of table.
- **Sixth rule:** if LDAP functionality is not enabled, routes calls to IP address 10.33.45.72

Therefore, once the device receives the incoming Tel call, the first rule that it uses is the fifth rule, which queries the AD server. When the AD replies, the device searches the table from the first rule down for the matching destination phone prefix (i.e., "PBX:", "OCS:", "MOBILE:", and "LDAP_ERR:"), and then sends the call to the appropriate destination.

9.6 Configuring DTMF Transport Types

You can control the way DTMF digits are transported over the IP network to the remote endpoint, by using one of the following modes:

- **Using INFO message according to Nortel IETF draft:** DTMF digits are carried to the remote side in INFO messages. To enable this mode, define the following:
  - RxDTMFOption = 0
  - TxDTMFOption = 1

  Note that in this mode, DTMF digits are erased from the audio stream (DTMFTransportType is automatically set to 0).
- **Using INFO message according to Cisco's mode:** DTMF digits are carried to the remote side in INFO messages. To enable this mode, define the following:
  - RxDTMFOption = 0
  - TxDTMFOption = 3

  Note that in this mode, DTMF digits are erased from the audio stream (DTMFTransportType is automatically set to 0).
- **Using NOTIFY messages according to <draft-mahy-sipping-signaled-digits-01.txt>**: DTMF digits are carried to the remote side using NOTIFY messages. To enable this mode, define the following:
  - RxDTMFOption = 0
  - TxDTMFOption = 2

  Note that in this mode, DTMF digits are erased from the audio stream (DTMFTransportType is automatically set to 0).
- **Using RFC 2833 relay with Payload type negotiation:** DTMF digits are carried to the remote side as part of the RTP stream in accordance with RFC 2833 standard. To enable this mode, define the following:
  - RxDTMFOption = 3
  - TxDTMFOption = 4

  Note that to set the RFC 2833 payload type with a different value (other than its default) configure the RFC2833PayloadType parameter. The device negotiates the RFC 2833 payload type using local and remote SDP and sends packets using the payload type from the received SDP. The device expects to receive RFC 2833 packets with the same payload type as configured by the RFC2833PayloadType parameter. If the remote side doesn't include 'telephony-event' in its SDP, the device sends DTMF digits in transparent mode (as part of the voice stream).
- **Sending DTMF digits (in RTP packets) as part of the audio stream (DTMF Relay is disabled):** This method is typically used with G.711 coders; with other low-bit rate (LBR) coders, the quality of the DTMF digits is reduced. To enable this mode, define the following:
  - RxDTMFOption = 0 (i.e., disabled)
  - TxDTMFOption = 0 (i.e., disabled)
  - DTMFTransportType = 2 (i.e., transparent)

- **Using INFO message according to Korea mode:** DTMF digits are carried to the remote side in INFO messages. To enable this mode, define the following:
  - RxDTMFOption = 0 (i.e., disabled)
  - TxDTMFOption = 3

  Note that in this mode, DTMF digits are erased from the audio stream (DTMFTransportType is automatically set to 0).

Notes:

- The device is always ready to receive DTMF packets over IP in all possible transport modes: INFO messages, NOTIFY, and RFC 2833 (in proper payload type) or as part of the audio stream.
- To exclude RFC 2833 Telephony event parameter from the device's SDP, set RxDTMFOption to 0 in the *ini* file.

The following parameters affect the way the device handles the DTMF digits:

- TxDTMFOption, RxDTMFOption, and RFC2833PayloadType
- MGCPDTMFDetectionPoint, DTMFVolume, DTMFTransportType, DTMFDigitLength, and DTMFInterDigitInterval

9.7 FXS and FXO Capabilities

9.7.1 FXS/FXO Coefficient Types

The FXS Coefficients and FXO Coefficients can be defined as one of the following types:

- US line type of 600 ohm AC impedance and 40 V RMS ringing voltage for REN = 2
- European standard (TBR21)

These types can be defined using the the *ini* file parameters FXSCountryCoefficients (for FXS) and CountryCoefficients (for FXO), or in the Web 'Analog Settings' page (refer to "Configuring the Analog Settings" on page 76).

These Coefficient types are used to increase return loss and trans-hybrid loss performance for two telephony line type interfaces (US or European). This adaptation is performed by modifying the telephony interface characteristics. This means, for example, that changing impedance matching or hybrid balance doesn't require hardware modifications, so that a single device is able to meet requirements for different markets. The digital design of the filters and gain stages also ensures high reliability, no drifts (over temperature or time) and simple variations between different line types.

The FXS Coefficient types provide best termination and transmission quality adaptation for two FXS line types interfaces. This parameter affects the following AC and DC interface parameters:

- DC (battery) feed characteristics
- AC impedance matching
- Transmit gain
- Receive gain
- Hybrid balance

- Frequency response in transmit and receive direction
- Hook thresholds
- Ringing generation and detection parameters

9.7.2 FXO Operating Modes

This section provides a description of the device's FXO operating modes:

- For IP-to-Tel calls (refer to "FXO Operations for IP-to-Tel Calls" on page 543)
- For Tel-to-IP calls (refer to "FXO Operations for Tel-to-IP Calls" on page 546)
- Call termination on FXO devices (refer to "Call Termination on FXO Devices" on page 548)

9.7.2.1 FXO Operations for IP-to-Tel Calls

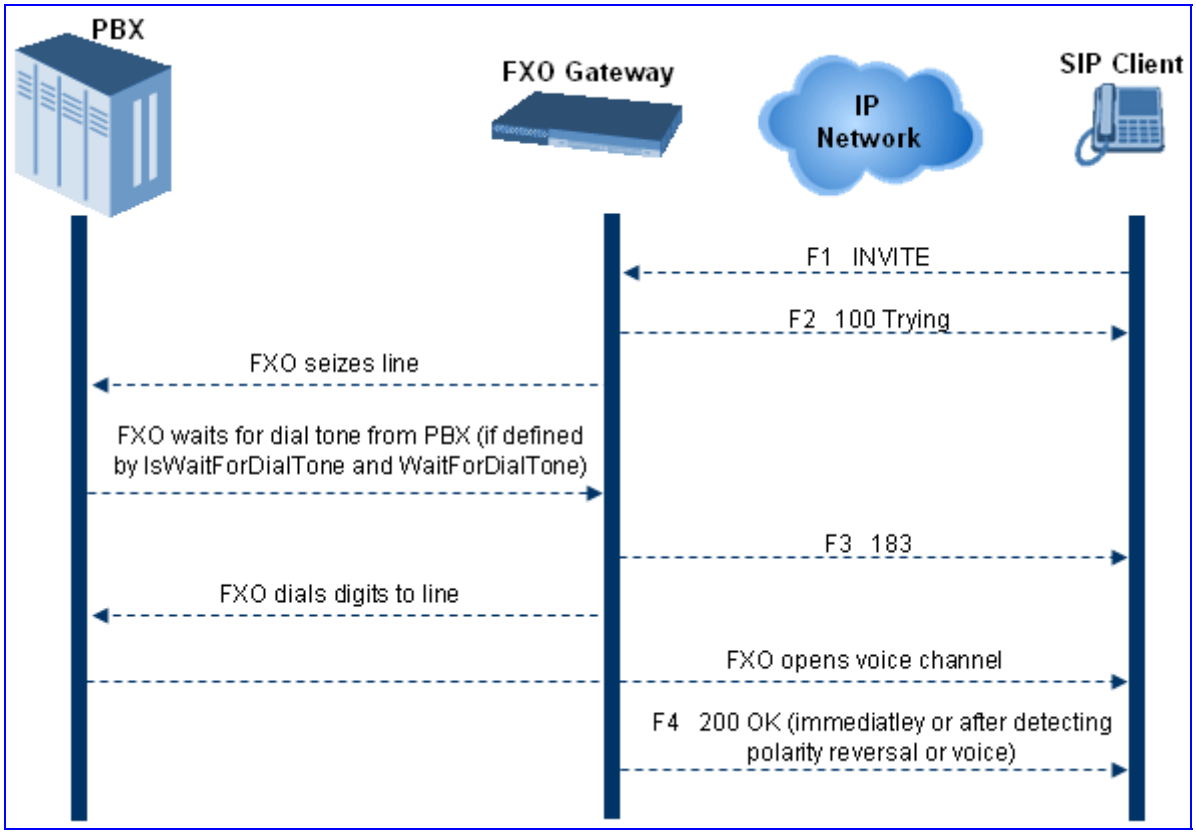
The FXO device provides the following operating modes for IP-to-Tel calls:

- One-stage dialing (refer to "One-Stage Dialing" on page 543)
  - Waiting for dial tone (refer to "Two-Stage Dialing" on page 545)
  - Time to wait before dialing
  - Answer supervision
- Two-stage dialing (refer to "Two-Stage Dialing" on page 545)
- Dialing time: DID wink (refer to "DID Wink" on page 545)

9.7.2.1.1 One-Stage Dialing

One-stage dialing is when the FXO device receives an IP-to-Tel call, off-hooks the PBX line connected to the telephone, and then immediately dials the destination telephone number. In other words, the IP caller doesn't dial the PSTN number upon hearing a dial tone.

Figure 9-41: Call Flow for One-Stage Dialing


PBX
FXO Gateway
IP Network
SIP Client
F1 INVITE
F2 100 Trying
FXO seizes line
FXO waits for dial tone from PBX (if defined by IsWaitForDialTone and WaitForDialTone)
F3 183
FXO dials digits to line
FXO opens voice channel
F4 200 OK (immediatley or after detecting polarity reversal or voice)

One-stage dialing incorporates the following FXO functionality:

- **Waiting for Dial Tone:** Enables the device to dial the digits to the Tel side only after detecting a dial tone from the PBX line. The *ini* file parameter IsWaitForDialTone is used to configure this operation.
- **Time to Wait Before Dialing:** Defines the time (in msec) between seizing the FXO line and starting to dial the digits. The *ini* file parameter WaitForDialTime is used to configure this operation.

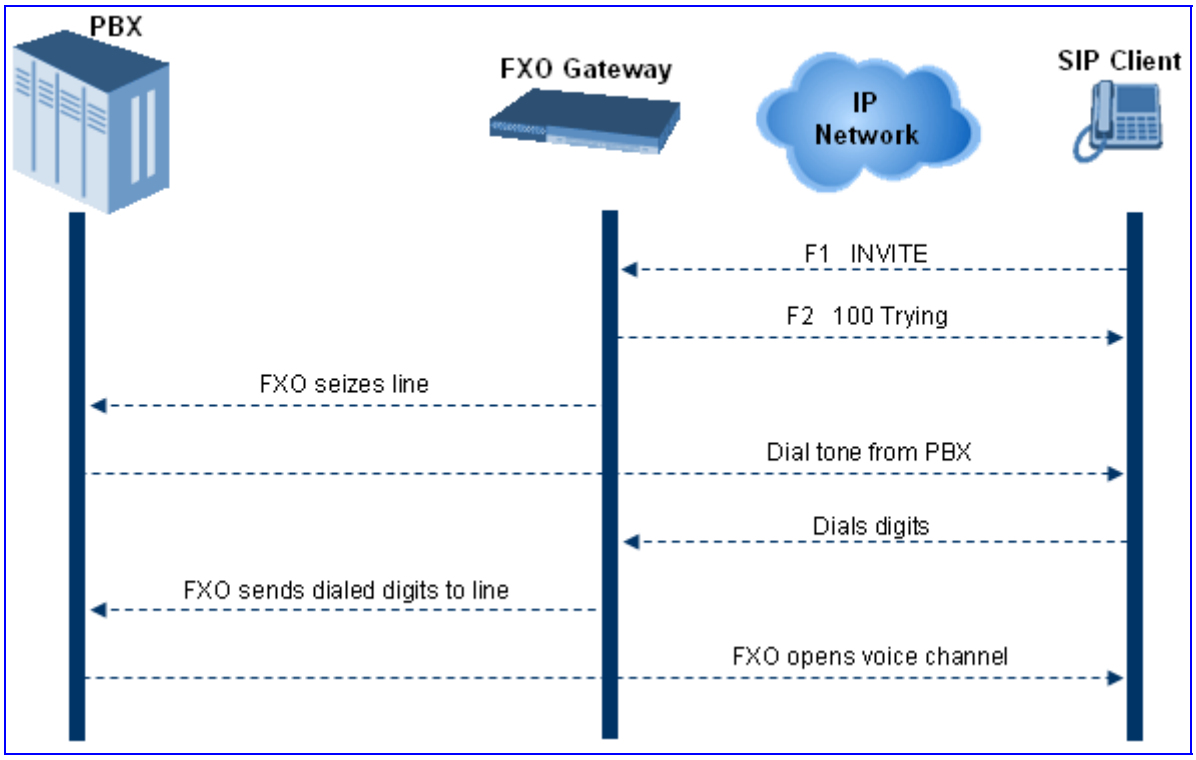
Note: The *ini* file parameter IsWaitForDialTone must be disabled for this mode.

- **Answer Supervision:** The Answer Supervision feature enables the FXO device to determine when a call is connected, by using one of the following methods:
  - Polarity Reversal: device sends a 200 OK in response to an INVITE only when it detects a polarity reversal.
  - Voice Detection: device sends a 200 OK in response to an INVITE only when it detects the start of speech (or ringback tone) from the Tel side. (Note that the IPM detectors must be enabled).

9.7.2.1.2 Two-Stage Dialing

Two-stage dialing is when the IP caller is required to dial twice. The caller initially dials to the FXO device and only after receiving a dial tone from the PBX (via the FXO device), dials the destination telephone number.

Figure 9-42: Call Flow for Two-Stage Dialing



Two-stage dialing implements the Dialing Time feature. Dialing Time allows you to define the time that each digit can be separately dialed. By default, the overall dialing time per digit is 200 msec. The longer the telephone number, the greater the dialing time.

The relevant parameters for configuring Dialing Time include the following:

- DTMFDigitLength (100 msec): time for generating DTMF tones to the PSTN (PBX) side
- DTMFInterDigitInterval (100 msec): time between generated DTMF digits to PSTN (PBX) side

9.7.2.1.3 DID Wink

The device's FXO ports support Direct Inward Dialing (DID). DID is a service offered by telephone companies that enables callers to dial directly to an extension on a PBX without the assistance of an operator or automated call attendant. This service makes use of DID trunks, which forward only the last three to five digits of a phone number to the PBX. If, for example, a company has a PBX with extensions 555-1000 to 555-1999, and a caller dials 555-1234, the local central office (CO) would forward, for example, only 234 to the PBX. The PBX would then ring extension 234.

DID wink enables the originating end to seize the line by going off-hook. It waits for acknowledgement from the other end before sending digits. This serves as an integrity check that identifies a malfunctioning trunk and allows the network to send a re-order tone to the calling party.

The "start dial" signal is a wink from the PBX to the FXO device. The FXO then sends the last four to five DTMF digits of the called number. The PBX uses these digits to complete the routing directly to an internal station (telephone or equivalent)

- DID Wink can be used for connection to EIA/TIA-464B DID Loop Start lines
- Both FXO (detection) and FXS (generation) are supported

9.7.2.2 FXO Operations for Tel-to-IP Calls

The FXO device provides the following FXO operating modes for Tel-to-IP calls:

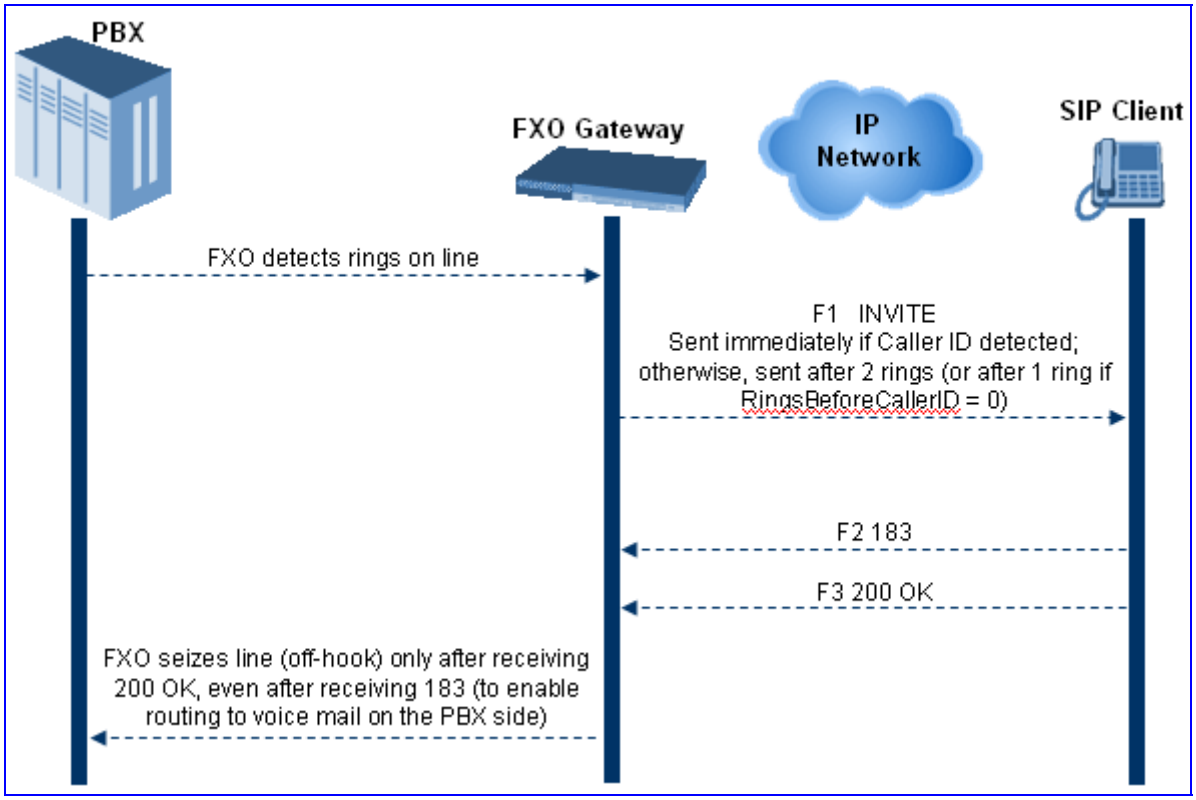
- Automatic Dialing (refer to "Automatic Dialing" on page 546)
- Collecting Digits Mode (refer to "Collecting Digits Mode" on page 547)
- FXO Supplementary Services (refer to "FXO Supplementary Services" on page 547)
  - Hold/Transfer Toward the Tel side
  - Hold/Transfer Toward the IP side
  - Blind Transfer to the Tel side

9.7.2.2.1 Automatic Dialing

Automatic dialing is defined using the *ini* file parameter table TargetOfChannel (refer to Analog Telephony Parameters) or the embedded Web server's 'Automatic Dialing' screen (refer to "Automatic Dialing" on page 172).

The SIP call flow diagram below illustrates Automatic Dialing.

Figure 9-43: Call Flow for Automatic Dialing

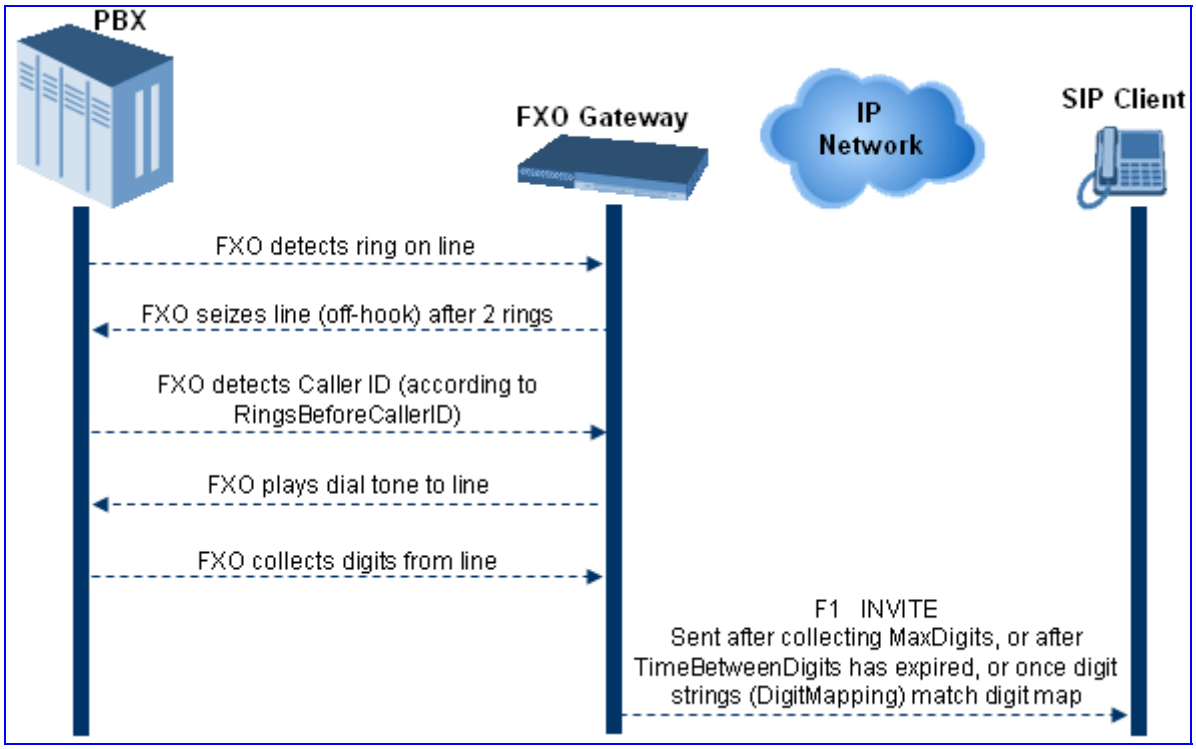

PBX
FXO Gateway
IP Network
SIP Client
FXO detects rings on line
F1 INVITE
Sent immediately if Caller ID detected; otherwise, sent after 2 rings (or after 1 ring if RingsBeforeCallerID = 0)
F2 183
F3 200 OK
FXO seizes line (off-hook) only after receiving 200 OK, even after receiving 183 (to enable routing to voice mail on the PBX side)

9.7.2.2.2 Collecting Digits Mode

When automatic dialing is not defined, the device collects the digits.

The SIP call flow diagram below illustrates the Collecting Digits Mode.

Figure 9-44: Call Flow for Collecting Digits Mode


PBX
FXO Gateway
IP Network
SIP Client
FXO detects ring on line
FXO seizes line (off-hook) after 2 rings
FXO detects Caller ID (according to RingsBeforeCallerID)
FXO plays dial tone to line
FXO collects digits from line
F1 INVITE
Sent after collecting MaxDigits, or after TimeBetweenDigits has expired, or once digit strings (DigitMapping) match digit map

9.7.2.2.3 FXO Supplementary Services

The FXO supplementary services include the following:

- **Hold / Transfer toward the Tel side:** The *ini* file parameter LineTransferMode must be set to 0 (default). If the FXO receives a hook-flash from the IP side (using out-of-band or RFC 2833), the device sends the hook-flash to the Tel side by performing one of the following:
  - Performing a hook flash (i.e., on-hook and off-hook)
  - Sending a hook-flash code (defined by the *ini* file parameter HookFlashCode)

  The PBX may generate a dial tone that is sent to the IP, and the IP side may dial digits of a new destination.
- **Blind Transfer to the Tel side:** A blind transfer is one in which the transferring phone connects the caller to a destination line before ringback begins. The *ini* file parameter LineTransferMode must be set to 1.

  The blind transfer call process is as follows:
  - FXO receives a REFER request from the IP side
  - FXO sends a hook-flash to the PBX, dials the digits (that are received in the Refer-To header), and then drops the line (on-hook). Note that the time between flash to dial is according to the WaitForDialTime parameter.

- PBX performs the transfer internally

- **Hold / Transfer toward the IP side:** The FXO device doesn't initiate hold / transfer as a response to input from the Tel side. If the FXO receives a REFER request (with or without replaces), it generates a new INVITE according to the Refer-To header.

9.7.2.3 Call Termination on FXO Devices

This section describes the device's call termination capabilities for its FXO interfaces:

- Calls terminated by a PBX (refer to "Call Termination by PBX" on page 548)
- Calls terminated before call establishment (refer to "Call Termination before Call Establishment" on page 549)
- Ring detection timeout (refer to "Ring Detection Timeout" on page 549)

9.7.2.3.1 Calls Termination by PBX

The FXO device supports various methods for identifying when a call has been terminated by the PBX.

The PBX doesn't disconnect calls, but instead signals to the device that the call has been disconnected using one of the following methods:

- **Detection of polarity reversal/current disconnect:** The call is immediately disconnected after polarity reversal or current disconnect is detected on the Tel side (assuming the PBX/CO generates this signal). This is the recommended method.

  Relevant parameters: EnableReversalPolarity, EnableCurrentDisconnect, CurrentDisconnectDuration, CurrentDisconnectDefaultThreshold, and TimeToSampleAnalogLineVoltage.

- **Detection of Reorder, Busy, Dial, and Special Information Tone (SIT) tones:** The call is immediately disconnected after a Reorder, Busy, Dial, or SIT tone is detected on the Tel side (assuming the PBX / CO generates this tone). This method requires the correct tone frequencies and cadence to be defined in the Call Progress Tones file. If these frequencies are not known, define them in the CPT file (the tone produced by the PBX / CO must be recorded and its frequencies analyzed -- refer to Adding a Reorder Tone to the CPT File in the Reference Manual). This method is slightly less reliable than the previous one. You can use the CPTWizard (described in the *Reference Manual*) to analyze Call Progress Tones generated by any PBX or telephone network.

  Relevant parameters: DisconnectOnBusyTone and DisconnectOnDialTone.

- **Detection of silence:** The call is disconnected after silence is detected on both call directions for a specific (configurable) amount of time. The call isn't disconnected immediately; therefore, this method should only be used as a backup option.

  Relevant parameters: EnableSilenceDisconnect and FarEndDisconnectSilencePeriod.

- **Special DTMF code:** A digit pattern that when received from the Tel side, indicates to the device to disconnect the call.

  Relevant *ini* file parameter: TelDisconnectCode.

- **Interruption of RTP stream:** Relevant parameters: BrokenConnectionEventTimeout and DisconnectOnBrokenConnection.

Note: This method operates correctly only if silence suppression is not used.

- **Protocol-based termination of the call from the IP side**

Note: The implemented disconnect method must be supported by the CO or PBX.

9.7.2.3.2 Call Termination before Call Establishment

The device supports the following call termination methods before a call is established:

- **Call termination upon receipt of SIP error response (in Automatic Dialing mode):** By default, when the FXO device operates in Automatic Dialing mode, there is no method to inform the PBX if a Tel-to-IP call has failed (SIP error response - 4xx, 5xx or 6xx - is received). The reason is that the FXO device does not seize the line until a SIP 200 OK response is received. Use the FXOAutoDialPlayBusyTone parameter to allow the device to play a Busy/Reorder tone to the PSTN line if a SIP error response is received. The FXO device seizes the line (off-hook) for the duration defined by the TimeForReorderTone parameter. After playing the tone, the line is released (on-hook).
- **Call termination after caller (PBX) on-hooks phone (Ring Detection Timeout feature):** This method operates in one of the following manners:
  - **Automatic Dialing is enabled:** if the remote IP party doesn't answer the call and the ringing signal (from the PBX) stops for a user-defined time (configured by the parameter FXOBetweenRingTime), the FXO device releases the IP call.
  - **No automatic dialing and Caller ID is enabled:** the device seizes the line after detection of the second ring signal (allowing detection of caller ID sent between the first and the second rings). If the second ring signal is not received within this timeout, the device doesn't initiate a call to IP.

9.7.2.3.3 Ring Detection Timeout

The operation of Ring Detection Timeout depends on the following:

- **Automatic dialing is disabled and Caller ID is enabled:** if the second ring signal is not received for a user-defined time (using the parameter FXOBetweenRingTime), the FXO device doesn't initiate a call to the IP.
- **Automatic dialing is enabled:** if the remote party doesn't answer the call and the ringing signal stops for a user-defined time (using the parameter FXOBetweenRingTime), the FXO device releases the IP call.

Ring Detection Timeout supports full ring cycle of ring on and ring off (from ring start to ring start).

9.7.3 Remote PBX Extension Between FXO and FXS Devices

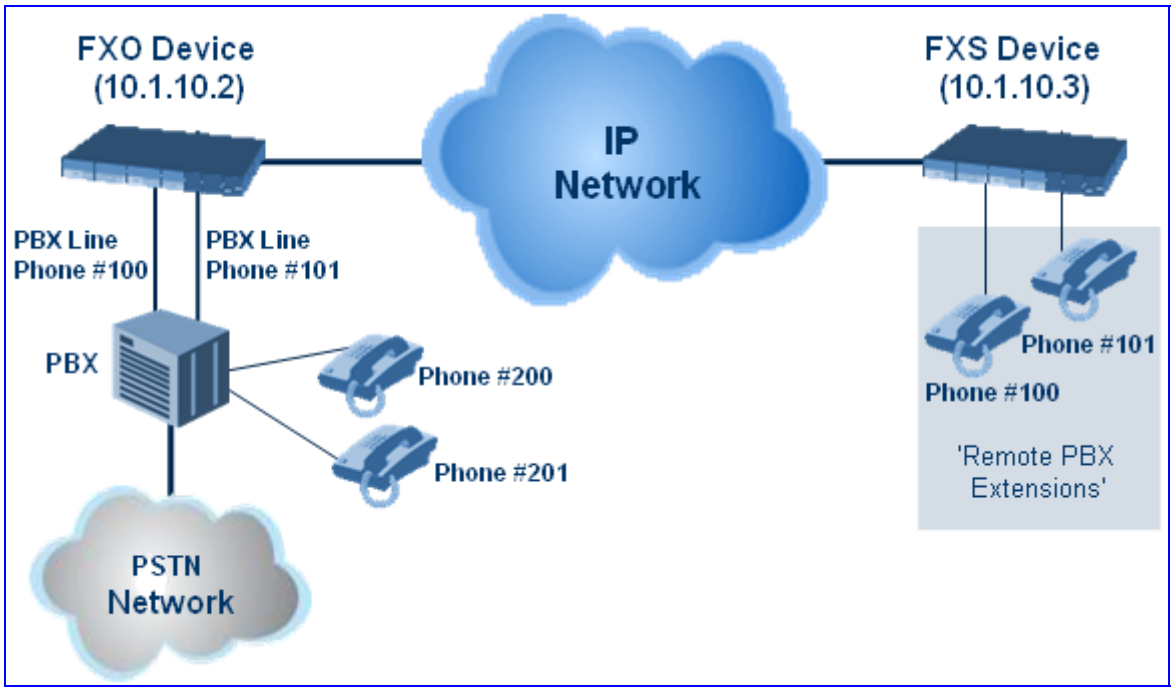
Remote PBX extension offers a company the capability of extending the "power" of its local PBX by allowing remote phones (remote offices) to connect to the company's PBX over the IP network (instead of via PSTN). This is as if the remote office is located in the head office (where the PBX is installed). PBX extensions are connected through FXO ports to the IP network, instead of being connected to individual telephone stations. At the remote office, FXS units connect analog phones to the same IP network. To produce full transparency, each FXO port is mapped to an FXS port (i.e., one-to-one mapping). This allows individual extensions to be extended to remote locations. To call a remote office worker, a PBX user or a PSTN caller simply dials the PBX extension that is mapped to the remote FXS port.

This section provides an example on how to implement a remote telephone extension through the IP network, using FXO andFXS interfaces. In this configuration, the FXO device routes calls received from the PBX to the 'Remote PBX Extension' connected to the FXS device. The routing is transparent as if the telephone connected to the FXS device is directly connected to the PBX.

The following is required:

- FXO interfaces with ports connected directly to the PBX lines (shown in the figure below)
- FXS interfaces for the 'remote PBX extension'
- Analog phones (POTS)
- PBX (one or more PBX loop start lines)
- LAN network

Figure 9-45: FXO-FXS Remote PBX Extension (Example)



9.7.3.1 Dialing from Remote Extension (Phone at FXS)

The procedure below describes how to dial from the 'remote PBX extension' (i.e., phone connected to the FXS interface).

➢ **To make a call from the FXS interface:**

1. Off-hook the phone and wait for the dial tone from the PBX. This is as if the phone is connected directly to the PBX. The FXS and FXO interfaces establish a voice path connection from the phone to the PBX immediately after the phone is off-hooked.
2. Dial the destination number (e.g., phone number 201). The DTMF digits are sent over IP directly to the PBX. All the audible tones are generated from the PBX (such as ringback, busy, or fast busy tones). One-to-one mapping occurs between the FXS ports and PBX lines.
3. The call disconnects when the phone connected to the FXS goes on-hook.

9.7.3.2 Dialing from PBX Line or PSTN

The procedure below describes how to dial from a PBX line (i.e., from a telephone directly connected to the PBX) or from the PSTN to the 'remote PBX extension' (i.e., telephone connected to the FXS interface).

➢ **To dial from a telephone directly connected to the PBX or from the PSTN:**

- Dial the PBX subscriber number (e.g., phone number 101) in the same way as if the user's phone was connected directly to the PBX. As soon as the PBX rings the FXO device, the ring signal is 'sent' to the phone connected to the FXS device. Once the phone connected to the FXS device is off-hooked, the FXO device seizes the PBX line and the voice path is established between the phone and PBX.

  There is one-to-one mapping between PBX lines and FXS device ports. Each PBX line is routed to the same phone (connected to the FXS device). The call disconnects when the phone connected to the FXS device is on-hooked.

9.7.3.3 Message Waiting Indication for Remote Extensions

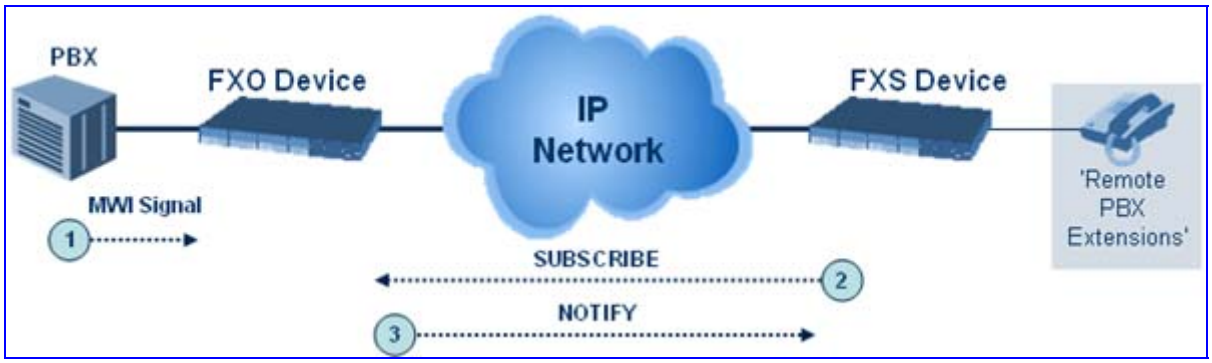
The device supports the relaying of Message Waiting Indications (MWI) for remote extensions (and voice mail applications). Instead of subscribing to an MWI server to receive notifications of pending messages, the FXO device receives subscriptions from the remote FXS device and notifies the appropriate extension when messages (and the number of messages) are pending.

The FXO device detects an MWI message from the Tel (PBX) side using any one of the following methods:

- 100 VDC (sent by the PBX to activate the phone's lamp)
- Stutter dial tone from the PBX
- MWI display signal (according to the parameter CallerIDType)

Upon detection of an MWI message, the FXO device sends a SIP NOTIFY message to the IP side. When receiving this NOTIFY message, the remote FXS device generates an MWI signal toward its Tel side.

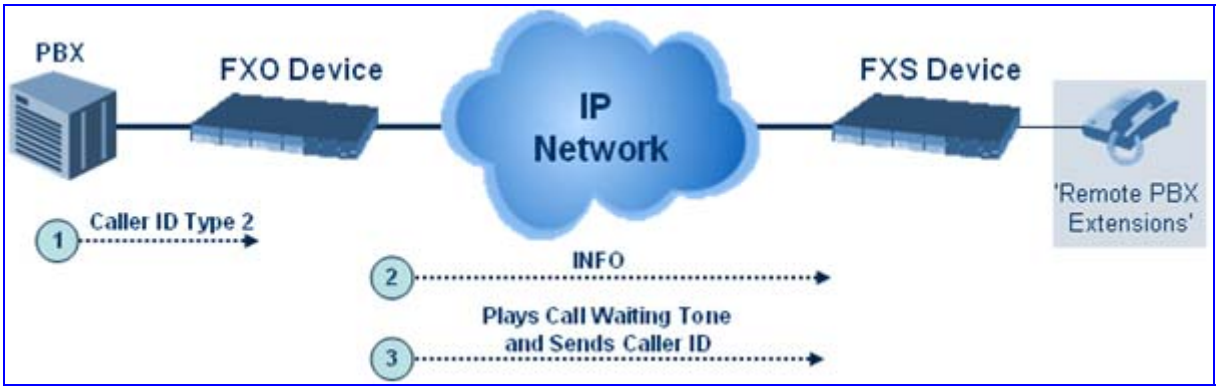
Figure 9-46: MWI for Remote Extensions



9.7.3.4 Call Waiting for Remote Extensions

When the FXO device detects a Call Waiting indication (FSK data of the Caller Id - CallerIDType2) from the PBX, it sends a proprietary INFO message, which includes the caller identification to the FXS device. Once the FXS device receives this INFO message, it plays a call waiting tone and sends the caller ID to the relevant port for display. The remote extension connected to the FXS device can toggle between calls using the Hook Flash button.

Figure 9-47: Call Waiting for Remote Extensions



9.7.3.5 FXS Gateway Configuration

The procedure below describes how to configure the FXS interface (at the 'remote PBX extension').

➢ **To configure the FXS interface:**

1. In the 'Trunk Group Table' page (refer to "Configuring the Trunk Group Table" on page 102, assign the phone numbers 100 to 104 to the device's endpoints.

Figure 9-48: Assigning Phone Numbers to FXS Endpoints

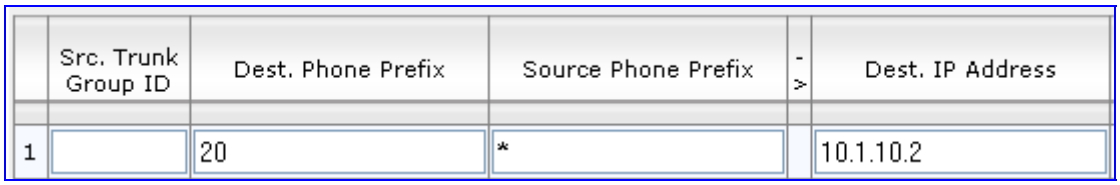
Group Index	Module	From Trunk	To Trunk	Channels	Phone Number	Trunk Group ID
1	Module 3 FXS	1	1	1-4	100	0

2. In the 'Automatic Dialing' page (refer to "Automatic Dialing" on page 172), enter the phone numbers of the FXO device in the 'Destination Phone Number' fields. When a phone connected to Port #1 off-hooks, the FXS device automatically dials the number '200'.

Figure 9-49: Automatic Dialing for FXS Ports

3. In the 'Outbound IP Routing Table' page (refer to "Configuring the Outbound IP Routing Table" on page 158), enter 20 for the destination phone prefix, and 10.1.10.2 for the IP address of the FXO device.

Figure 9-50: FXS Tel-to-IP Routing Configuration



Note: For the transfer to function in remote PBX extensions, Hold must be disabled at the FXS device (i.e., Enable Hold = 0) and hook-flash must be transferred from the FXS to the FXO (HookFlashOption = 4).

9.7.3.6 FXO Gateway Configuration

The procedure below describes how to configure the FXO interface (to which the PBX is directly connected).

➢ **To configure the FXO interface:**

1. In the 'Trunk Group Table' page (refer to "Configuring the Trunk Group Table" on page 102, assign the phone numbers 200 to 204 to the device's FXO endpoints.

Figure 9-51: Assigning Phone Numbers to FXO Ports

2. In the 'Automatic Dialing' page, enter the phone numbers of the FXS device in the 'Destination Phone Number' fields. When a ringing signal is detected at Port #1, the FXO device automatically dials the number '100'.

Figure 9-52: FXO Automatic Dialing Configuration

Gateway Port	Destination Phone Number	Auto Dial Status
Module 3 Port 1 FXO	100	Enable
Module 3 Port 2 FXO	101	Enable
Module 3 Port 3 FXO	102	Enable
Module 3 Port 4 FXO	103	Enable

3. In the 'Outbound IP Routing Table' page, enter 10 in the 'Destination Phone Prefix' field, and the IP address of the FXS device (10.1.10.3) in the field 'IP Address'.

Figure 9-53: FXO Tel-to-IP Routing Configuration

	Dest. Phone Prefix	Source Phone Prefix	->	Dest. IP Address
1	10	*		10.1.10.3

4. In the 'FXO Settings' page (refer to "Configuring the FXO Parameters" on page 187), set the parameter 'Dialing Mode' to 'Two Stages' (IsTwoStageDial = 1).

9.8 Configuring Alternative Routing (Based on Connectivity and QoS)

The Alternative Routing feature enables reliable routing of Tel-to-IP calls when a Proxy isn't used. The device periodically checks the availability of connectivity and suitable Quality of Service (QoS) before routing. If the expected quality cannot be achieved, an alternative IP route for the prefix (phone number) is selected.

The following parameters are used to configure the Alternative Routing mechanism:

- AltRoutingTel2IPEnable
- AltRoutingTel2IPMode
- IPConnQoSMaxAllowedPL
- IPConnQoSMaxAllowedDelay

Note: If the alternative routing destination is the device itself, the call can be configured to be routed back to one of the device's Trunk Groups and thus, back into the PSTN (PSTN Fallback).

9.8.1 Alternative Routing Mechanism

When the device routes a Tel-to-IP call, the destination number is compared to the list of prefixes defined in the 'Outbound IP Routing Table' (described in "Configuring the Outbound IP Routing Table" on page 158). This table is scanned for the destination number's prefix starting at the top of the table. For this reason, you must enter the main IP route above any alternative route in the table. When an appropriate entry (destination number matches one of the prefixes) is found, the prefix's corresponding destination IP address is verified. If the destination IP address is disallowed (or if the original call fails and the device has made two additional attempts to establish the call without success), an alternative route is searched in the table and used for routing the call.

Destination IP address is disallowed if no ping to the destination is available (ping is continuously initiated every seven seconds), when an inappropriate level of QoS was detected or when a DNS host name is not resolved. The QoS level is calculated according to delay or packet loss of previously ended calls. If no call statistics are received for two minutes, the QoS information is reset.

9.8.2 Determining the Availability of Destination IP Addresses

To determine the availability of each destination IP address (or host name) in the routing table, one or all of the following user-defined methods are applied:

- Connectivity: The destination IP address is queried periodically (currently only by ping).
- **QoS:** The QoS of an IP connection is determined according to RTCP statistics of previous calls. Network delay (in msec) and network packet loss (in percentage) are separately quantified and compared to a certain (configurable) threshold. If the calculated amounts (of delay or packet loss) exceed these thresholds, the IP connection is disallowed.
- **DNS resolution:** When host name is used (instead of IP address) for the destination route, it is resolved to an IP address by a DNS server. Connectivity and QoS are then applied to the resolved IP address.

9.8.3 PSTN Fallback

The PSTN Fallback feature enables the device to redirect PSTN originated calls back to the legacy PSTN network if a destination IP route is unsuitable (disallowed) for voice traffic at a specific time. To enable PSTN fallback, assign the device's IP address as an alternative route to the desired prefixes. Note that calls (now referred to as IP-to-Tel calls) can be re-routed to a specific Trunk Group using the Routing parameters (refer to "Configuring the Inbound IP Routing Table" on page 163).

9.9 Fax and Modem Capabilities

This section describes the device's fax and modem capabilities, and includes the following main subsections:

- Fax and modem operating modes (refer to "Fax/Modem Operating Modes" on page 556)
- Fax and modem transport modes (refer to "Fax/Modem Transport Modes" on page 556)
- V.34 fax support (refer to "V.34 Fax Support" on page 562)
- V.152 support (refer to "V.152 Support" on page 563)

9.9.1 Fax/Modem Operating Modes

The device supports two modes of operation:

- Fax/modem negotiation that is not performed during the establishment of the call.
- Voice-band data (VBD) mode for V.152 implementation (refer to "V.152 Support" on page 563): fax/modem capabilities are negotiated between the device and the remote endpoint at the establishment of the call. During a call, when a fax/modem signal is detected, transition from voice to VBD (or T.38) is automatically performed and no additional SIP signaling is required. If negotiation fails (i.e., no match is achieved for any of the transport capabilities), fallback to existing logic occurs (according to the parameter IsFaxUsed).

9.9.2 Fax/Modem Transport Modes

The device supports the following transport modes for fax per modem type (V.22/V.23/Bell/V.32/V.34):

- T.38 fax relay (refer to "T.38 Fax Relay Mode" on page 557)
- G.711 Transport: switching to G.711 when fax/modem is detected (refer to "G.711 Fax / Modem Transport Mode" on page 558)
- Fax fallback to G.711 if T.38 is not supported (refer to "Fax Fallback" on page 558)
- Fax and modem bypass: a proprietary method that uses a high bit rate coder (refer to "Fax/Modem Bypass Mode" on page 559)
- NSE Cisco's Pass-through bypass mode for fax and modem (refer to "Fax / Modem NSE Mode" on page 560)
- Transparent with events: passing the fax / modem signal in the current voice coder with adaptations (refer to "Fax / Modem Transparent with Events Mode" on page 561)
- Transparent: passing the fax / modem signal in the current voice coder (refer to "Fax / Modem Transparent Mode" on page 561)
- RFC 2833 ANS Report upon Fax/Modem Detection (refer to "RFC 2833 ANS Report upon Fax/Modem Detection" on page 562)

'Adaptations' refer to automatic reconfiguration of certain DSP features for handling fax/modem streams differently than voice.

9.9.2.1 T.38 Fax Relay Mode

In Fax Relay mode, fax signals are transferred using the T.38 protocol. T.38 is an ITU standard for sending fax across IP networks in real-time mode. The device currently supports only the T.38 UDP syntax.

T.38 can be configured in the following ways:

- Switching to T.38 mode using SIP Re-INVITE messages (refer to "Switching to T.38 Mode using SIP Re-INVITE" on page 557)
- Automatically switching to T.38 mode without using SIP Re-INVITE messages (refer to "Automatically Switching to T.38 Mode without SIP Re-INVITE" on page 558)

When fax transmission ends, the reverse switching from fax relay to voice is automatically performed at both the local and remote endpoints.

You can change the fax rate declared in the SDP, using the parameter FaxRelayMaxRate (this parameter doesn't affect the actual transmission rate). In addition, you can enable or disable Error Correction Mode (ECM) fax mode using the FaxRelayECMEnable parameter.

When using T.38 mode, you can define a redundancy feature to improve fax transmission over congested IP networks. This feature is activated using the FaxRelayRedundancyDepth and FaxRelayEnhancedRedundancyDepth parameters. Although this is a proprietary redundancy scheme, it should not create problems when working with other T.38 decoders.

9.9.2.1.1 Switching to T.38 Mode using SIP Re-INVITE

In the Switching to T.38 Mode using SIP Re-INVITE mode, upon detection of a fax signal the terminating device negotiates T.38 capabilities using a Re-INVITE message. If the far-end device doesn't support T.38, the fax fails. In this mode, the parameter FaxTransportMode is ignored.

To configure T.38 mode using SIP Re-INVITE messages, set IsFaxUsed to 1. Additional configuration parameters include the following:

- FaxRelayEnhancedRedundancyDepth
- FaxRelayRedundancyDepth
- FaxRelayECMEnable
- FaxRelayMaxRate

Note: The terminating gateway sends T.38 packets immediately after the T.38 capabilities are negotiated in SIP. However, the originating device by default, sends T.38 (assuming the T.38 capabilities are negotiated in SIP) only after it receives T.38 packets from the remote device. This default behavior cannot be used when the originating device is located behind a firewall that blocks incoming T.38 packets on ports that have not yet received T.38 packets from the internal network. To resolve this problem, the device should be configured to send CNG packets in T.38 upon CNG signal detection (CNGDetectorMode = 1).

9.9.2.1.2 Automatically Switching to T.38 Mode without SIP Re-INVITE

In the Automatically Switching to T.38 Mode without SIP Re-INVITE mode, when a fax signal is detected, the channel automatically switches from the current voice coder to answer tone mode, and then to T.38-compliant fax relay mode.

To configure automatic T.38 mode, perform the following configurations:

- IsFaxUsed = 0
- FaxTransportMode = 1
- Additional configuration parameters:
  - FaxRelayEnhancedRedundancyDepth
  - FaxRelayRedundancyDepth
  - FaxRelayECMEnable
  - FaxRelayMaxRate

9.9.2.2 G.711 Fax / Modem Transport Mode

In this mode, when the terminating device detects fax or modem signals (CED or AnsAM), it sends a Re-INVITE message to the originating device requesting it to re-open the channel in G.711 VBD with the following adaptations:

- Echo Canceller = off
- Silence Compression = off
- Echo Canceller Non-Linear Processor Mode = off
- Dynamic Jitter Buffer Minimum Delay = 40
- Dynamic Jitter Buffer Optimization Factor = 13

After a few seconds upon detection of fax V.21 preamble or super G3 fax signals, the device sends a second Re-INVITE enabling the echo canceller (the echo canceller is disabled only on modem transmission).

A 'gpmd' attribute is added to the SDP according to the following format:

- **For G.711A-law:** a=gpmd:0 vbd=yes;ecan=on (or off, for modems)
- **For G.711 µ-law:** a=gpmd:8 vbd=yes;ecan=on (or off for modems)

The parameters FaxTransportMode and VxxModemTransportType are ignored and automatically set to the mode called 'transparent with events'.

To configure fax / modem transparent mode, set IsFaxUsed to 2.

9.9.2.3 Fax Fallback

In this mode, when the terminating device detects a fax signal, it sends a Re-INVITE message to the originating device with T.38. If the remote device doesn't support T.38 (replies with SIP response 415 'Media Not Supported'), the device sends a new Re-INVITE with G.711 VBD with the following adaptations:

- Echo Canceller = on
- Silence Compression = off
- Echo Canceller Non-Linear Processor Mode = off

- Dynamic Jitter Buffer Minimum Delay = 40
- Dynamic Jitter Buffer Optimization Factor = 13

When the device initiates a fax session using G.711, a 'gpmd' attribute is added to the SDP according to the following format:

- **For G.711A-law:** a=gpmd:0 vbd=yes;ecan=on
- **For G.711 µ-law:** a=gpmd:8 vbd=yes;ecan=on

In this mode, the parameter FaxTransportMode is ignored and automatically set to 'transparent'.

To configure fax fallback mode, set IsFaxUsed to 3.

9.9.2.4 Fax/Modem Bypass Mode

In this proprietary mode, when fax or modem signals are detected, the channel automatically switches from the current voice coder to a high bit-rate coder (according to the parameter FaxModemBypassCoderType). In addition, the channel is automatically reconfigured with the following fax / modem adaptations:

- Disables silence suppression
- Enables echo cancellation for fax
- Disables echo cancellation for modem
- Performs certain jitter buffering optimizations

The network packets generated and received during the bypass period are regular voice RTP packets (per the selected bypass coder), but with a different RTP payload type (according to the parameters FaxBypassPayloadType and ModemBypassPayloadType). During the bypass period, the coder uses the packing factor, which is defined by the parameter FaxModemBypassM. The packing factor determines the number of coder payloads (each the size of FaxModemBypassBasicRTPPacketInterval) that are used to generate a single fax/modem bypass packet. When fax/modem transmission ends, the reverse switching, from bypass coder to regular voice coder is performed.

To configure fax / modem bypass mode, perform the following configurations:

- IsFaxUsed = 0
- FaxTransportMode = 2
- V21ModemTransportType = 2
- V22ModemTransportType = 2
- V23ModemTransportType = 2
- V32ModemTransportType = 2
- V34ModemTransportType = 2
- BellModemTransportType = 2
- Additional configuration parameters:
  - FaxModemBypassCoderType
  - FaxBypassPayloadType
  - ModemBypassPayloadType

- FaxModemBypassBasicRTPPacketInterval
- FaxModemBypassDJBufMinDelay

Note: When the device is configured for modem bypass and T.38 fax, V.21 low-speed modems are not supported and fail as a result.

Tip: When the remote (non-AudioCodes') gateway uses G711 coder for voice and doesn't change the coder payload type for fax or modem transmission, it is recommended to use the Bypass mode with the following configuration:

- EnableFaxModemInbandNetworkDetection = 1
- FaxModemBypassCoderType = same coder used for voice
- FaxModemBypassM = same interval as voice
- ModemBypassPayloadType = 8 if voice coder is A-Law; 0 if voice coder is Mu-Law

9.9.2.5 Fax / Modem NSE Mode

In this mode, fax and modem signals are transferred using Cisco-compatible Pass-through bypass mode. Upon detection of fax or modem answering tone signal, the terminating device sends three to six special NSE RTP packets (using NSEpayloadType, usually 100). These packets signal the remote device to switch to G.711 coder (according to the parameter FaxModemBypassCoderType). After a few NSE packets are exchanged between the devices, both devices start using G.711 packets with standard payload type (8 for G.711 A-Law and 0 for G.711 Mu-Law). In this mode, no Re-INVITE messages are sent. The voice channel is optimized for fax/modem transmission (same as for usual bypass mode).

The parameters defining payload type for the proprietary AudioCodes' Bypass mode FaxBypassPayloadType and ModemBypassPayloadType are not used with NSE Bypass.

When configured for NSE mode, the device includes in its SDP the following line:

```
a=rtpmap:100 X-NSE/8000
```

(where 100 is the NSE payload type)

The Cisco gateway must include the following definition: "modem passthrough nse payload-type 100 codec g711alaw".

To configure NSE mode, perform the following configurations:

- IsFaxUsed = 0
- FaxTransportMode = 2
- NSEMode = 1
- NSEPayloadType = 100
- V21ModemTransportType = 2
- V22ModemTransportType = 2
- V23ModemTransportType = 2
- V32ModemTransportType = 2

- V34ModemTransportType = 2
- BellModemTransportType = 2

9.9.2.6 Fax / Modem Transparent with Events Mode

In this mode, fax and modem signals are transferred using the current voice coder with the following automatic adaptations:

- Echo Canceller = on (or off, for modems)
- Echo Canceller Non-Linear Processor Mode = off
- Jitter buffering optimizations

To configure fax / modem transparent with events mode, perform the following configurations:

- IsFaxUsed = 0
- FaxTransportMode = 3
- V21ModemTransportType = 3
- V22ModemTransportType = 3
- V23ModemTransportType = 3
- V32ModemTransportType = 3
- V34ModemTransportType = 3
- BellModemTransportType = 3

9.9.2.7 Fax / Modem Transparent Mode

In this mode, fax and modem signals are transferred using the current voice coder without notifications to the user and without automatic adaptations. It's possible to use the Profiles mechanism (refer to "Coders and Profile Definitions" on page 127) to apply certain adaptations to the channel used for fax / modem (e.g., to use the coder G.711, to set the jitter buffer optimization factor to 13, and to enable echo cancellation for fax and disable it for modem).

To configure fax / modem transparent mode, use the following parameters:

- IsFaxUsed = 0
- FaxTransportMode = 0
- V21ModemTransportType = 0
- V22ModemTransportType = 0
- V23ModemTransportType = 0
- V32ModemTransportType = 0
- V34ModemTransportType = 0
- BellModemTransportType = 0
- Additional configuration parameters:
  - CodersGroup

- DJBufOptFactor
- EnableSilenceCompression
- EnableEchoCanceller

Note: This mode can be used for fax, but is not recommended for modem transmission. Instead, use the modes Bypass (refer to "Fax/Modem Bypass Mode" on page 559) or Transparent with Events (refer to "Fax / Modem Transparent with Events Mode" on page 561) for modem.

9.9.2.8 RFC 2833 ANS Report upon Fax/Modem Detection

The device (terminator gateway) sends RFC 2833 ANS/ANSam events upon detection of fax and/or modem answer tones (i.e., CED tone). This causes the originator to switch to fax/modem. This parameter is applicable only when the fax or modem transport type is set to bypass, Transparent-with-Events, V.152 VBD, or G.711 transport. When the device is located on the originator side, it ignores these RFC 2833 events

Relevant parameters:

- IsFaxUsed = 0 or 3
- FaxTransportType = 2
- FaxModemNTEMode = 1
- VxxModemTransportType = 2

9.9.3 V.34 Fax Support

V.34 fax machines can transmit data over IP to the remote side using various methods. The device supports the following modes for transporting V.34 fax data over IP:

- Bypass mechanism for V.34 fax transmission (refer to "Using Bypass Mechanism for V.34 Fax Transmission" on page 562)
- T38 Version 0 relay mode, i.e., fallback to T.38 (refer to "Using Relay mode for both T.30 and V.34 faxes" on page 563)

Using the *ini* file parameter V34FaxTransportType, you can configure whether to pass V.34 over T38 fax relay, or use Bypass over the High Bit Rate coder (e.g. PCM A-Law).

Note: The CNG detector is disabled (CNGDetectorMode = 0) in all the subsequent examples.

9.9.3.1 Using Bypass Mechanism for V.34 Fax Transmission

In this proprietary scenario, the device uses bypass (or NSE) mode to transmit V.34 faxes, enabling the full utilization of its speed.

Configure the following parameters to use bypass mode for both T.30 and V.34 faxes:

- FaxTransportMode = 2 (Bypass)
- V34ModemTransportType = 2 (Modem bypass)

- V32ModemTransportType = 2
- V23ModemTransportType = 2
- V22ModemTransportType = 2

Configure the following parameters to use bypass mode for V.34 faxes and T.38 for T.30 faxes:

- FaxTransportMode = 1 (Relay)
- V34ModemTransportType = 2 (Modem bypass)
- V32ModemTransportType = 2
- V23ModemTransportType = 2
- V22ModemTransportType = 2

9.9.3.2 Using Relay mode for both T.30 and V.34 faxes

In this scenario, V.34 fax machines are forced to use their backward compatibility with T.30 faxes and operate in the slower T.30 mode.

Use the following parameters to use T.38 mode for both V.34 and T.30 faxes:

- FaxTransportMode = 1 (Relay)
- V34ModemTransportType = 0 (Transparent)
- V32ModemTransportType = 0
- V23ModemTransportType = 0
- V22ModemTransportType = 0

9.9.4 V.152 Support

The device supports the ITU-T recommendation V.152 (Procedures for Supporting Voice-Band Data over IP Networks). Voice-band data (VBD) is the transport of modem, facsimile, and text telephony signals over a voice channel of a packet network with a codec appropriate for such signals.

For V.152 capability, the device supports T.38 as well as VBD codecs (i.e., G.711 A-law and G.711 µ-law). The selection of capabilities is performed using the coders table (refer to "Configuring Coders" on page 128).

When in VBD mode for V.152 implementation, support is negotiated between the device and the remote endpoint at the establishment of the call. During this time, initial exchange of call capabilities is exchanged in the outgoing SDP. These capabilities include whether VBD is supported and associated RTP payload types ('gpmd' SDP attribute), supported codecs, and packetization periods for all codec payload types ('ptime' SDP attribute). After this initial negotiation, no Re-INVITE messages are necessary as both endpoints are synchronized in terms of the other side's capabilities. If negotiation fails (i.e., no match was achieved for any of the transport capabilities), fallback to existing logic occurs (according to the parameter IsFaxUsed).

Below is an example of media descriptions of an SDP indicating support for V.152.

```
v=0
o=-  0 0 IN IPV4 <IPAdressA>
s=-
t=0 0
p=+1
c=IN IP4  <IPAddressA
m=audio <udpPort A> RTP/AVP 18 0
a=ptime:10
a=rtpmap:96 PCMU/8000
a=gpmd: 96 vbd=yes
```

In the example above, V.152 implementation is supported (using the dynamic payload type 96 and G.711 u-law as the VBD codec) as well as the voice codecs G.711 µ-law and G.729.

Instead of using VBD transport mode, the V.152 implementation can use alternative relay fax transport methods (e.g., fax relay over IP using T.38). The preferred V.152 transport method is indicated by the SDP 'pmft' attribute. Omission of this attribute in the SDP content means that VBD mode is the preferred transport mechanism for voice-band data.

To configure T.38 mode, use the CodersGroup parameter.

9.10 Working with Supplementary Services

The device supports the following supplementary services:

- Call Hold and Retrieve (refer to "Call Hold and Retrieve" on page 564)
- Consultation (refer to Consultation Feature on page 567)
- Call Transfer (refer to "Call Transfer" on page 567)
- Call Forward (refer to "Call Forward" on page 568)
- Call Waiting (refer to Call Waiting on page 571)
- Message Waiting Indication (refer to "Message Waiting Indication" on page 572)
- Caller ID (refer to Caller ID on page 573)
- Three-way conferencing (refer to Three-Way Conferencing on page 576)

To activate these supplementary services, enable each service's corresponding parameter using the Web interface or *ini* file.

Notes:

- All call participants must support the specific supplementary service that is used.
- When working with certain application servers (such as BroadSoft's BroadWorks) in client server mode (the application server controls all supplementary services and keypad features by itself), the device's supplementary services must be disabled.

9.10.1 Call Hold and Retrieve

Initiating Call Hold and Retrieve:

- Active calls can be put on-hold by pressing the phone's hook-flash button.
- The party that initiates the hold is called the *holding* party; the other party is called the *held* party.
- After a successful Hold, the holding party hears a Dial tone (HELD_TONE defined in the device's Call Progress Tones file).
- Call retrieve can be performed only by the holding party while the call is held and active.
- The holding party performs the retrieve by pressing the telephone's hook-flash button.
- After a successful retrieve, the voice is connected again.
- Hold is performed by sending a Re-INVITE message with IP address 0.0.0.0 or a=sendonly in the SDP according to the parameter HoldFormat.
- The hold and retrieve functionalities are implemented by Re-INVITE messages. The IP address 0.0.0.0 as the connection IP address or the string 'a=inactive' in the received Re-INVITE SDP cause the device to enter Hold state and to play the Held tone (configured in the device) to the PBX/PSTN. If the string 'a=sendonly' is received in the SDP message, the device stops sending RTP packets, but continues to listen to the incoming RTP packets. Usually, the remote party plays, in this scenario, Music on Hold (MOH) and the device forwards the MOH to the held party.

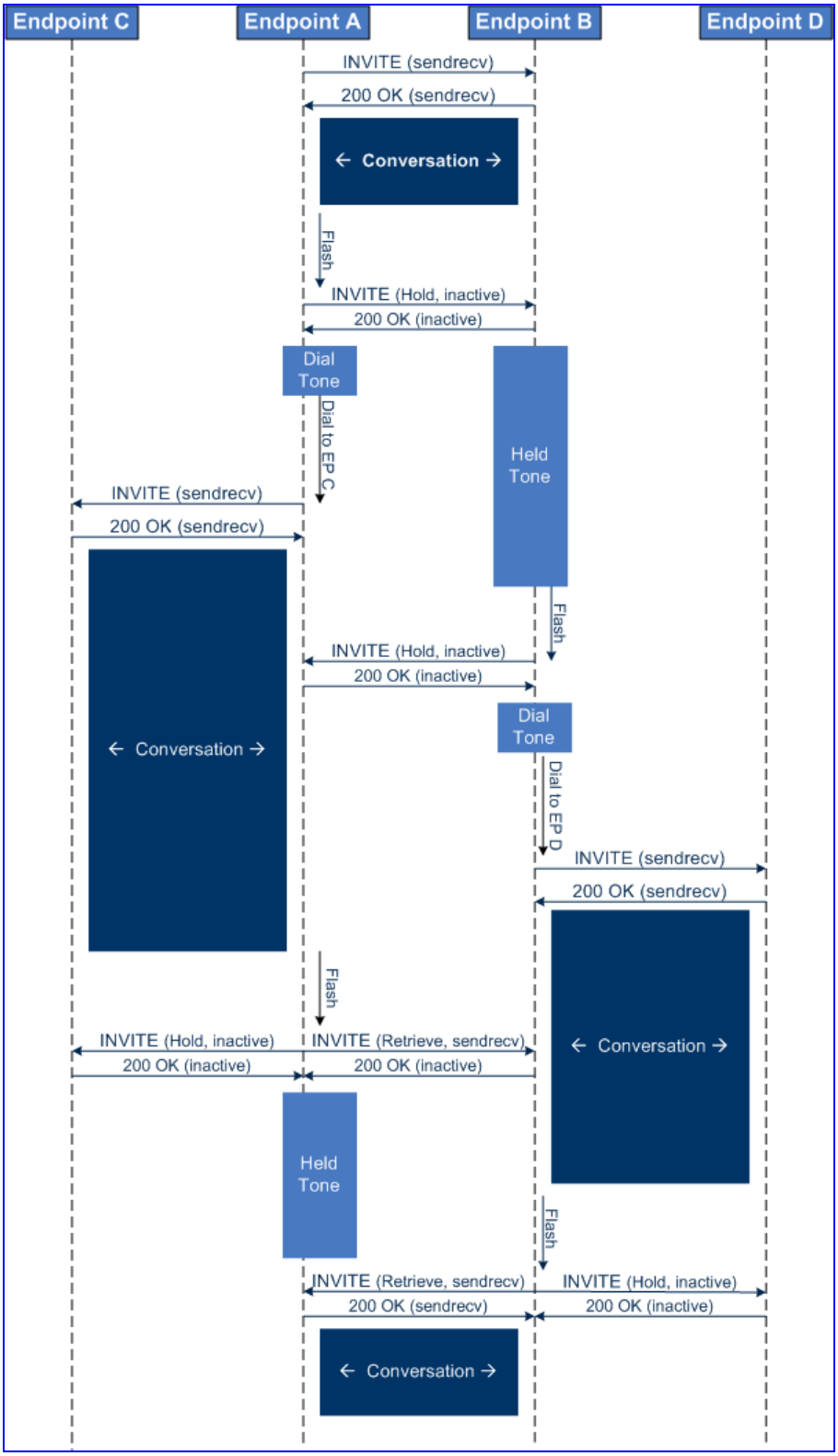
Receiving Hold/Retrieve:

- When an active call receives a Re-INVITE message with either the IP address 0.0.0.0 or the 'inactive' string in SDP, the device stops sending RTP and plays a local Held tone.
- When an active call receives a Re-INVITE message with the 'sendonly' string in SDP, the device stops sending RTP and listens to the remote party. In this mode, it is expected that on-hold music (or any other hold tone) is played (over IP) by the remote party.

You can also configure the device to keep a call on-hold for a user-defined time after which the call is disconnected, using the *ini* file parameter HeldTimeout.

The device also supports "double call hold" for FXS interfaces where the called party, which has been placed on-hold by the calling party, can then place the calling party on hold as well and make a call to another destination. The flowchart below provides an example of this type of call hold:

Figure 9-54: Double Hold SIP Call Flow



The flowchart above describes the following "double" call-hold scenario:

1. A calls B and establishes a voice path.
2. A places B on hold; A hears a Dial tone and B hears a Held tone.
3. A calls C and establishes a voice path.
4. B places A on hold; B hears a Dial tone.
5. B calls D and establishes a voice path.
6. A ends call with C; A hears a Held tone.
7. B ends call with D.
8. B retrieves call with A.

Notes:

- If a party that is placed on hold (e.g., B in the above example) is called by another party (e.g., D), then the on-hold party receives a Call Waiting tone instead of the Held tone.
- While in a Double Hold state, placing the phone on-hook disconnects both calls (i.e. call transfer is not performed).

9.10.2 Consultation Feature

The device's Consultation feature allows you to place one number on hold and consult privately with another party.

- The Consultation feature is relevant only for the holding party.
- After holding a call (by pressing hook-flash), the holding party hears a dial tone and can then initiate a new call, which is called a Consultation call.
- While hearing a dial tone, or when dialing to the new destination (before dialing is complete), the user can retrieve the held call by pressing hook-flash.
- The held call can't be retrieved while Ringback tone is heard.
- After the Consultation call is connected, the user can toggle between the held and active call by pressing the hook-flash key.

Note: The Consultation feature is applicable only to FXS interfaces.

9.10.3 Call Transfer

There are two types of call transfers:

- **Consultation Transfer** (REFER and REPLACES)**:** The common method to perform a consultation transfer is as follows:

  In the transfer scenario there are three parties - Party A = transferring, Party B = transferred, Party C = transferred to.

  1. A Calls B.
  2. B answers.
  3. A presses the hook-flash button and places B on-hold (party B hears a hold tone).
  4. A dials C.
  5. After A completes dialing C, A can perform the transfer by on-hooking the A phone.
  6. After the transfer is complete, B and C parties are engaged in a call.

  The transfer can be initiated at any of the following stages of the call between A and C:

  - Just after completing dialing C phone number - transfer from setup.
  - While hearing Ringback – transfer from alert.
  - While speaking to C - transfer from active.

- **Blind Transfer** (REFER)**:** Blind transfer is performed after a call is established between A and B, and party A decides to immediately transfer the call to C without speaking with C. The result of the transfer is a call between B and C (similar to consultation transfer, but skipping the consultation stage).

Notes:

- Call transfer is initiated by sending REFER with REPLACES.
- The device can receive and act upon receiving REFER with or without REPLACES.
- The device can receive and act upon receiving INVITE with REPLACES, in which case the old call is replaced by the new one.
- The INVITE with REPLACES can be used to implement Directed Call Pickup.

9.10.4 Call Forward

For digital interfaces: The device supports Call Deflection (ETS-300-207-1) for Euro ISDN and QSIG (ETSI TS 102 393) for Network and User sides, which provides IP-ISDN interworking of call forwarding (call diversion) when the device receives a SIP 302 response.

Call forward performed by the SIP side: Upon receipt of a Facility message with Call Rerouting IE from the PSTN, the device initiates a SIP transfer process by sending a SIP 302 (including the Call Rerouting destination number) to the IP in response to the remote SIP entity's INVITE message. The device then responds with a Disconnect message to the PSTN side.

Call forward performed by the PSTN side: When the device sends the INVITE message to the remote SIP entity and receives a SIP 302 response, the device sends a Facility message with the same IE mentioned above to the PSTN, and waits for the PSTN side to disconnect the call. This is configured using the CallReroutingMode.

For analog interfaces: The following methods of call forwarding are supported:

- **Immediate:** incoming call is forwarded immediately and unconditionally.
- **Busy:** incoming call is forwarded if the endpoint is busy.
- **No Reply:** incoming call is forwarded if it isn't answered for a specified time.
- **On Busy or No Reply:** incoming call is forwarded if the port is busy or when calls are not answered after a specified time.
- **Do Not Disturb:** immediately reject incoming calls. Upon receiving a call for a Do Not Disturb, the 603 Decline SIP response code is sent.

Three forms of forwarding parties are available:

- **Served party:** party configured to forward the call (FXS device).
- **Originating party:** party that initiates the first call (FXS or FXO device).
- **Diverted party:** new destination of the forwarded call (FXS or FXO device).

The served party (FXS interface) can be configured through the Web interface (refer to "Configuring Call Forward" on page 175) or *ini* file to activate one of the call forward modes. These modes are configurable per endpoint.

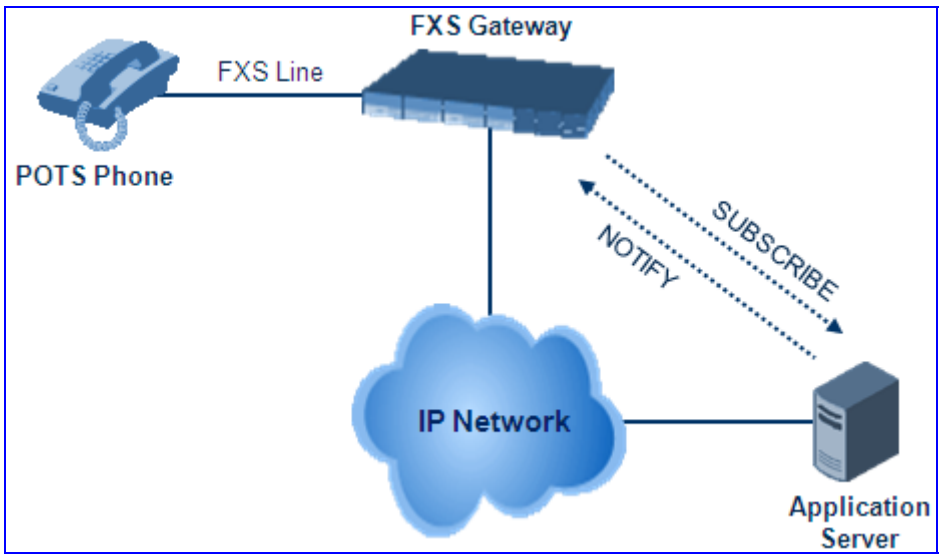
Notes:

- When call forward is initiated, the device sends a SIP 302 response with a contact that contains the phone number from the forward table and its corresponding IP address from the routing table (or when a proxy is used, the proxy's IP address).
- For receiving call forward, the device handles SIP 3xx responses for redirecting calls with a new contact.

9.10.4.1 Call Forward Reminder Ring

The device supports the Call Forward Reminder Ring feature for FXS interfaces, whereby the device's FXS endpoint emits a short ring burst (only if in onhook state) when a third-party Application Server (e.g., softswitch) forwards an incoming call to another destination. This is important in that it notifies (audibly) the FXS endpoint user that a call forwarding service is currently being performed.

Figure 9-55: Call Forward Reminder with Application Server



The device generates a Call Forward Reminder ring burst to the FXS endpoint each time it receives a SIP NOTIFY message with a “reminder ring” xml body. The NOTIFY request is sent from the Application Server to the device each time the Application Server forwards an incoming call. The service is cancelled when an UNSUBSCRIBE request is sent from the device, or when the Subscription time expires.

The Reminder Ring tone can be defined by using the parameter CallForwardRingToneID, which points to a ring tone defined in the Call Progress Tone file.

The following parameters are used to configure this feature:

- EnableNRTSubscription
- ASSubscribeIPGroupID
- NRTRetrySubscriptionTime
- CallForwardRingToneID

9.10.4.2 Call Forward Reminder (Off-Hook) Special Dial Tone

The device plays a special dial tone (Stutter Dial tone - Tone Type #15) to a specific FXS endpoint when the phone is off-hooked and when a third-party Application server (AS), e.g., a softswitch is used to forward calls intended for the endpoint, to another destination. This is useful in that it reminds the FXS user of this service. This feature does not involve device subscription (SIP SUBSCRIBE) to the AS.

Activation/deactivation of the service is notified by the server. An unsolicited SIP NOTIFY request is sent from the AS to the device when the Call Forward service is activated or cancelled. Depending on this NOTIFY request, the device plays either the standard dial tone or the special dial tone for Call Forward.

For playing the special dial tone, the received SIP NOTIFY message must contain the following headers:

- **From and To:** contain the same information, indicating the specific endpoint
- **Event:** ua-profile
- **Content-Type:** "application/simservs+xml"
- Message body is the XML body and contains the “dial-tone-pattern” set to "special-condition-tone" (<ss:dial-tone-pattern>special-condition-tone</ss:dial-tone-pattern>), which is the special tone indication.

For cancelling the special dial tone and playing the regular dial tone, the received SIP NOTIFY message must contain the following headers:

- **From and To:** contain the same information, indicating the specific endpoint
- **Event:** ua-profile
- **Content-Type:** "application/simservs+xml"
- Message body is the XML body containing the “dial-tone-pattern” set to "standard-condition-tone" (<ss:dial-tone-pattern>standard-condition-tone</ss:dial-tone-pattern>), which is the regular dial tone indication.

Therefore, the special dial tone is valid until another SIP NOTIFY is received that instructs otherwise (as described above).

Note: if the MWI service is active, the MWI dial tone overrides this special Call Forward dial tone

9.10.5 Call Waiting

The Call Waiting feature enables FXS devices to accept an additional (second) call on busy endpoints. If an incoming IP call is designated to a busy port, the called party hears a call waiting tone (several configurable short beeps) and (for Bellcore and ETSI Caller IDs) can view the Caller ID string of the incoming call. The calling party hears a Call Waiting Ringback Tone. The called party can accept the new call using hook-flash, and can toggle between the two calls.

➢ **To enable call waiting:**

1. Set the parameter EnableCallWaiting to 1.
2. Set the parameter EnableHold to 1.
3. Define the Call Waiting indication and Call Waiting Ringback tones in the Call Progress Tones file. You can define up to four Call Waiting indication tones (refer to the parameter FirstCallWaitingToneID in "SIP Configuration Parameters" on page 284).
4. To configure the Call Waiting indication tone cadence, modify the following parameters: NumberOfWaitingIndications, WaitingBeepDuration and TimeBetweenWaitingIndications.

5. To configure a delay interval before a Call Waiting Indication is played to the currently busy port, use the parameter TimeBeforeWaitingIndication. This enables the caller to hang up before disturbing the called party with Call Waiting Indications. Applicable only to FXS modules.

Both the calling and called sides are supported by FXS interfaces; FXO interfaces support only the calling side.

To indicate Call Waiting, the device sends a 182 Call Queued response. The device identifies Call Waiting when a 182 Call Queued response is received.

Note: The Call Waiting feature is applicable only to FXS/FXO interfaces.

9.10.6 Message Waiting Indication

For analog interfaces: The device supports Message Waiting Indication (MWI) according to IETF <draft-ietf-sipping-mwi-04.txt>, including SUBSCRIBE (to MWI server). The FXS device can accept an MWI NOTIFY message that indicates waiting messages or that the MWI is cleared. Users are informed of these messages by a stutter dial tone. The stutter and confirmation tones are defined in the CPT file (refer to the *Product Reference Manual*). If the MWI display is configured, the number of waiting messages is also displayed. If the MWI lamp is configured, the phone's lamp (on a phone that is equipped with an MWI lamp) is lit. The device can subscribe to the MWI server per port (usually used on FXS) or per device (used on FXO).

To configure MWI, use the following parameters:

- EnableMWI
- MWIServerIP
- MWIAnalogLamp
- MWIDisplay
- StutterToneDuration
- EnableMWISubscription
- MWIExpirationTime
- SubscribeRetryTime
- SubscriptionMode
- CallerIDType (determines the standard for detection of MWI signals)
- ETSIVMWITypeOneStandard
- BellcoreVMWITypeOneStandard

For Digital PSTN interfaces: The device supports the interworking of QSIG Message Waiting Indication (MWI) to IP. This provides interworking between an ISDN PBX with voicemail capabilities and a softswitch, which requires information on the number of messages waiting for a specific user. This support is configured using the MWIInterrogationType parameter. This parameter determines the device's handling of MWI Interrogation messages.

The process for sending the MWI status upon request from a softswitch is as follows:

1. The softswitch sends a SIP SUBSCRIBE message to the device.
2. The device responds by sending an empty SIP NOTIFY to the softswitch, and then sending an ISDN Setup message with Facility IE containing an MWI Interrogation request to the PBX.
3. The PBX responds by sending to the device an ISDN Connect message containing Facility IE with an MWI Interrogation result, which includes the number of voice messages waiting for the specific user.
4. The device sends another SIP NOTIFY to the softswitch, containing this MWI information.
5. The SIP NOTIFY messages are sent to the IP Group defined by the NotificationIPGroupID parameter.

In addition, when a change in the status occurs (e.g., a new voice message is waiting or the user has retrieved a message from the voice mail), the PBX initiates an ISDN Setup message with Facility IE containing an MWI Activate request, which includes the new number of voice messages waiting for the user. The device forwards this information to the softswitch by sending a SIP NOTIFY.

Depending on the PBX support, the MWIInterrogationType parameter can be configured to handle these MWI Interrogation messages in different ways. For example, some PBXs support only the MWI Activate request (and not MWI Interrogation request). Some support both these requests. Therefore, the device can be configured to disable this feature, or enable it with one of the following support:

- Responds to MWI Activate requests from the PBX by sending SIP NOTIFY MWI messages (i.e., does not send MWI Interrogation messages).
- Send MWI Interrogation message, but don't use its result. Instead, wait for MWI Activate requests from the PBX.
- Send MWI Interrogation message, use its result, and use the MWI Activate requests.

9.10.7 Caller ID

This section discusses the device's Caller ID support.

Note: The Caller ID feature is applicable only to FXS/FXO interfaces.

9.10.7.1 Caller ID Detection / Generation on the Tel Side

By default, generation and detection of Caller ID to the Tel side is disabled. To enable Caller ID, set the parameter EnableCallerID to 1. When the Caller ID service is enabled:

- For FXS: the Caller ID signal is sent to the device's port
- For FXO: the Caller ID signal is detected

The configuration for Caller ID is described below:

- Use the parameter CallerIDType to define the Caller ID standard. Note that the Caller ID standard that is used on the PBX or phone must match the standard defined in the device.
- Select the Bellcore caller ID sub standard using the parameter BellcoreCallerIDTypeOneSubStandard
- Select the ETSI FSK caller ID sub standard using the parameter ETSICallerIDTypeOneSubStandard
- Enable or disable (per port) the caller ID generation (for FXS) and detection (for FXO) using the 'Generate / Detect Caller ID to Tel' table (EnableCallerID). If a port isn't configured, its caller ID generation / detection are determined according to the global parameter EnableCallerID.
- EnableCallerIDTypeTwo: disables / enables the generation of Caller ID type 2 when the phone is off-hooked (used for call waiting).
- RingsBeforeCallerID: sets the number of rings before the device starts detection of caller ID (FXO only). By default, the device detects the caller ID signal between the first and second rings.
- AnalogCallerIDTimimgMode: determines the time period when a caller ID signal is generated (FXS only). By default, the caller ID is generated between the first two rings.
- PolarityReversalType: some Caller ID signals use reversal polarity and/or wink signals. In these scenarios, it is recommended to set PolarityReversalType to 1 (Hard) (FXS only).
- The Caller ID interworking can be changed using the parameters UseSourceNumberAsDisplayName and UseDisplayNameAsSourceNumber.

9.10.7.2 Debugging a Caller ID Detection on FXO

The procedure below describes debugging caller ID detection in FXO interfaces.

➢ **To debug a Caller ID detection on an FXO interface:**

1. Verify that the parameter EnableCallerID is set to 1.
2. Verify that the caller ID standard (and substandard) of the device matches the standard of the PBX (using the parameters CallerIDType, BellcoreCallerIDTypeOneSubStandard, and ETSICallerIDTypeOneSubStandard).
3. Define the number of rings before the device starts the detection of caller ID (using the parameter RingsBeforeCallerID).
4. Verify that the correct FXO coefficient type is selected (using the parameter CountryCoefficients), as the device is unable to recognize caller ID signals that are distorted.
5. Connect a phone to the analog line of the PBX (instead of to the device's FXO interface) and verify that it displays the caller ID.

If the above does not solve the problem, you need to record the caller ID signal (and send it to AudioCodes), as described below.

➢ **To record the caller ID signal using the debug recording mechanism:**

1. Access the FAE page (by appending "FAE" to the device's IP address in the Web browser's URL, for example, http://10.13.4.13/FAE).
2. Press the **Cmd Shell** link.
3. Enter the following commands:

```
dr
ait <IP address of PC to collect the debug traces sent from
the device>
AddChannelIdTrace ALL-WITH-PCM <port number, which starts from
0>
Start
```

4. Make a call to the FXO.
5. To stop the DR recording, at the CLI prompt, type **STOP**.

9.10.7.3 Caller ID on the IP Side

Caller ID is provided by the SIP From header containing the caller's name and "number", for example:

```
From: "David" <SIP:101@10.33.2.2>;tag=35dfsgasd45dg
```

If Caller ID is restricted (received from Tel or configured in the device), the From header is set to:

```
From: "anonymous" <anonymous@anonymous.invalid>; tag=35dfsgasd45dg
```

The P-Asserted (or P-Preferred) headers are used to present the originating party's caller ID even when the caller ID is restricted. These headers are used together with the Privacy header.

- If Caller ID is restricted:
  - The From header is set to "anonymous" <anonymous@anonymous.invalid>
  - The 'Privacy: id' header is included
  - The P-Asserted-Identity (or P-Preferred-Identity) header shows the caller ID
- If Caller ID is allowed:
  - The From header shows the caller ID
  - The 'Privacy: none' header is included
  - The P-Asserted-Identity (or P-Preferred-Identity) header shows the caller ID

In addition, the caller ID (and presentation) can be displayed in the Calling Remote-Party-ID header.

The 'Caller Display Information' table (CallerDisplayInfo) is used for the following:

- **FXS interfaces** - to define the caller ID (per port) that is sent to IP.
- **FXO interfaces** - to define the caller ID (per port) that is sent to IP if caller ID isn't detected on the Tel side, or when EnableCallerID = 0.
- **FXS and FXO interfaces** - to determine the presentation of the caller ID (allowed or restricted).
- **To maintain backward compatibility** - when the strings 'Private' or 'Anonymous' are set in the Caller ID/Name field, the caller ID is restricted and the value in the Presentation field is ignored.

The value of the 'Presentation' field that is defined in the 'Caller Display Information' table can be overridden by configuring the 'Presentation' parameter in the 'Tel to IP Source Number Manipulation' table. Therefore, this table can be used to set the presentation for specific calls according to Source / Destination prefixes.

The caller ID can be restricted/allowed (per port) using keypad features KeyCLIR and KeyCLIRDeact (FXS only).

AssertedIdMode defines the header that is used (in the generated INVITE request) to deliver the caller ID (P-Asserted-Identity or P-preferred-Identity). Use the parameter UseTelURIForAssertedID to determine the format of the URI in these headers (sip: or tel:).

The parameter EnableRPIheader enables Remote-Party-ID (RPI) headers for calling and called numbers for Tel-to-IP calls.

9.10.8 Three-Way Conferencing

The device supports three-way conference calls. These conference calls can also occur simultaneously.

The following example demonstrates three-way conferencing. This example assumes that a telephone "A" connected to the device wants to establish a three-way conference call with two remote IP phones "B" and "C":

1. User A has an ongoing call with IP phone B.
2. User A places IP phone B on hold (by pressing the telephone's flash hook button, defined by the parameter HookFlashCode).
3. User A hears a dial tone, and then makes a call to IP phone C.
4. IP phone C answers the call.
5. User A can now establish a three-way conference call (between A, B and C) by pressing the flash-hook button, defined by the parameter ConferenceCode (e.g., regular flash-hook button or "*1").

Notes:

- Instead of using the flash-hook button to establish a three-way conference call, you can dial a user-defined hook-flash code (e.g., "*1"), configured by the parameter HookFlashCode.
- Three-way conferencing is applicable only to FXS interfaces.

The device supports the following conference modes (configured by the parameter 3WayConferenceMode):

- **Conferencing controlled by an external AudioCodes Conference (media) server:** The Conference-initiating INVITE sent by the device uses the ConferenceID concatenated with a unique identifier as the Request-URI. This same Request-URI is set as the Refer-To header value in the REFER messages that are sent to the two remote parties. For this mode, the parameter 3WayConferenceMode is set to 0 (default.)
- **Conferencing controlled by an external, third-party Conference (media) server:** The Conference-initiating INVITE sent by the device uses only the ConferenceID as the Request-URI. The Conference server sets the Contact header of the 200 OK response to the actual unique identifier (Conference URI) to be used by the participants. This Conference URI is included (by the device) in the Refer-To header value in the REFER messages sent by the device to the remote parties. The remote parties join the conference by sending INVITE messages to the Conference server using this conference URI. For this mode, the parameter 3WayConferenceMode is set to 1.

To enable three-way conferencing, the following parameters need to be configured:

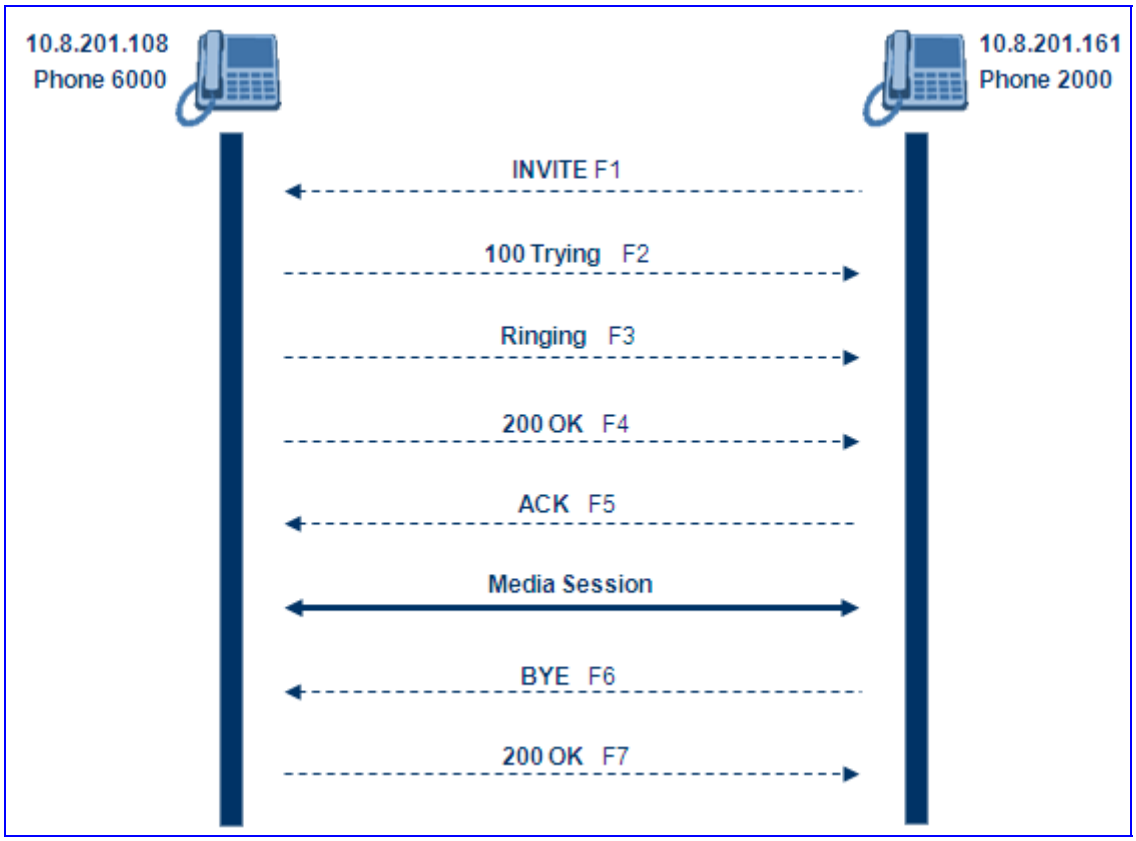
- Enable3WayConference
- ConferenceCode = '!' (default, which is the hook flash button)
- HookFlashCode
- 3WayConferenceMode (conference mode)

9.11 Routing Examples

9.11.1 SIP Call Flow Example

The SIP call flow (shown in the following figure), describes SIP messages exchanged between two devices during a basic call. In this call flow example, device (10.8.201.158) with phone number '6000' dials device (10.8.201.161) with phone number '2000'.

Figure 9-56: SIP Call Flow



- **F1 INVITE (10.8.201.108 >> 10.8.201.161):**

```
INVITE sip:2000@10.8.201.161;user=phone SIP/2.0
Via: SIP/2.0/UDP 10.8.201.108;branch=z9hG4bKacsiJkDGd
From: <sip:6000@10.8.201.108>;tag=1c5354
To: <sip:2000@10.8.201.161>
Call-ID: 534366556655skKw-6000--2000@10.8.201.108
CSeq: 18153 INVITE
Contact: <sip:8000@10.8.201.108;user=phone>
User-Agent: Audiocodes-Sip-Gateway/Mediant 1000/v.6.00.010.006
Supported: 100rel,em
Allow: REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,
NOTIFY,PRACK,REFER,INFO
Content-Type: application/sdp
Content-Length: 208

v=0
o=AudiocodesGW 18132 74003 IN IP4 10.8.201.108
s=Phone-Call
c=IN IP4 10.8.201.108
t=0 0
m=audio 4000 RTP/AVP 8 96
a=rtpmap:8 pcma/8000
```

```
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
a=ptime:20
```

- **F2 TRYING (10.8.201.161 >> 10.8.201.108):**

```
SIP/2.0 100 Trying
Via: SIP/2.0/UDP 10.8.201.108;branch=z9hG4bKacsiJkDGd
From: <sip:6000@10.8.201.108>;tag=1c5354
To: <sip:2000@10.8.201.161>
Call-ID: 534366556655skKw-6000--2000@10.8.201.108
Server: Audiocodes-Sip-Gateway/Mediant 1000/v.6.00.010.006
CSeq: 18153 INVITE
Content-Length: 0
```

- **F3 RINGING 180 (10.8.201.161 >> 10.8.201.108):**

```
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP 10.8.201.108;branch=z9hG4bKacsiJkDGd
From: <sip:6000@10.8.201.108>;tag=1c5354
To: <sip:2000@10.8.201.161>;tag=1c7345
Call-ID: 534366556655skKw-6000--2000@10.8.201.108
Server: Audiocodes-Sip-Gateway/Mediant 1000/v.6.00.010.006
CSeq: 18153 INVITE
Supported: 100rel,em
Content-Length: 0
```

Note: Phone '2000' answers the call and then sends a 200 OK message to device 10.8.201.108.

- **F4 200 OK (10.8.201.161 >> 10.8.201.108):**

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.8.201.108;branch=z9hG4bKacsiJkDGd
From: <sip:6000@10.8.201.108>;tag=1c5354
To: <sip:2000@10.8.201.161>;tag=1c7345
Call-ID: 534366556655skKw-6000--2000@10.8.201.108
CSeq: 18153 INVITE
Contact: <sip:2000@10.8.201.161;user=phone>
Server: Audiocodes-Sip-Gateway/Mediant 1000/v.6.00.010.006
Supported: 100rel,em
Allow: REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,
NOTIFY,PRACK,REFER,INFO
Content-Type: application/sdp
Content-Length: 206

v=0
o=AudiocodesGW 30221 87035 IN IP4 10.8.201.161
s=Phone-Call
c=IN IP4 10.8.201.10
t=0 0
m=audio 7210 RTP/AVP 8 96
a=rtpmap:8 pcma/8000
a=ptime:20
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
```

- **F5 ACK (10.8.201.108 >> 10.8.201.10):**

```
ACK sip:2000@10.8.201.161;user=phone SIP/2.0
Via: SIP/2.0/UDP 10.8.201.108;branch=z9hG4bKacZYpJWxZ
From: <sip:6000@10.8.201.108>;tag=1c5354
To: <sip:2000@10.8.201.161>;tag=1c7345
Call-ID: 534366556655skKw-6000--2000@10.8.201.108
User-Agent: Audiocodes-Sip-Gateway/Mediant 1000/v.6.00.010.006
CSeq: 18153 ACK
Supported: 100rel,em
Content-Length: 0
```

Note: Phone '6000' goes on-hook and device 10.8.201.108 sends a BYE to device 10.8.201.161. A voice path is established.

- **F6 BYE (10.8.201.108 >> 10.8.201.10):**

```
BYE sip:2000@10.8.201.161;user=phone SIP/2.0
Via: SIP/2.0/UDP 10.8.201.108;branch=z9hG4bKacRKCVBud
From: <sip:6000@10.8.201.108>;tag=1c5354
To: <sip:2000@10.8.201.161>;tag=1c7345
Call-ID: 534366556655skKw-6000--2000@10.8.201.108
User-Agent: Audiocodes-Sip-Gateway/Mediant 1000/v.6.00.010.006
CSeq: 18154 BYE
Supported: 100rel,em
Content-Length: 0
```

- **F7 OK 200 (10.8.201.10 >> 10.8.201.108):**

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.8.201.108;branch=z9hG4bKacRKCVBud
From: <sip:6000@10.8.201.108>;tag=1c5354
To: <sip:2000@10.8.201.161>;tag=1c7345
Call-ID: 534366556655skKw-6000--2000@10.8.201.108
Server: Audiocodes-Sip-Gateway/Mediant 1000/v.6.00.010.006
CSeq: 18154 BYE
Supported: 100rel,em
Content-Length: 0
```

9.11.2 SIP Authentication Example

The device supports basic and digest (MD5) authentication types, according to SIP RFC 3261 standard. A proxy server might require authentication before forwarding an INVITE message. A Registrar/Proxy server may also require authentication for client registration. A proxy replies to an unauthenticated INVITE with a 407 Proxy Authorization Required response, containing a Proxy-Authenticate header with the form of the challenge. After sending an ACK for the 407, the user agent can then re-send the INVITE with a Proxy-Authorization header containing the credentials.

User agents, Redirect or Registrar servers typically use 401 Unauthorized response to challenge authentication containing a WWW-Authenticate header, and expect the re-INVITE to contain an Authorization header.

The following example describes the Digest Authentication procedure, including computation of user agent credentials:

1. The REGISTER request is sent to a Registrar/Proxy server for registration:

```
REGISTER sip:10.2.2.222 SIP/2.0
Via: SIP/2.0/UDP 10.1.1.200
From: <sip: 122@10.1.1.200>;tag=1c17940
To: <sip: 122@10.1.1.200>
Call-ID: 634293194@10.1.1.200
User-Agent: Audiocodes-Sip-Gateway/Mediant 1000/v.6.00.010.006
CSeq: 1 REGISTER
Contact: sip:122@10.1.1.200:
Expires:3600
```

2. Upon receipt of this request, the Registrar/Proxy returns a 401 Unauthorized response:

```
SIP/2.0 401 Unauthorized
Via: SIP/2.0/UDP 10.2.1.200
From: <sip:122@10.2.2.222 >;tag=1c17940
To: <sip:122@10.2.2.222 >
Call-ID: 634293194@10.1.1.200
Cseq: 1 REGISTER
Date: Mon, 30 Jul 2001 15:33:54 GMT
Server: Columbia-SIP-Server/1.17
Content-Length: 0
WWW-Authenticate: Digest realm="audiocodes.com",
nonce="11432d6bce58ddf02e3b5e1c77c010d2",
stale=FALSE,
algorithm=MD5
```

3. According to the sub-header present in the WWW-Authenticate header, the correct REGISTER request is created.
4. Since the algorithm is MD5:
   - The username is equal to the endpoint phone number 122.
   - The realm return by the proxy is audiocodes.com.
   - The password from the *ini* file is AudioCodes.
   - The equation to be evaluated is (according to RFC this part is called A1) **'122:audiocodes.com:AudioCodes'.**
   - The MD5 algorithm is run on this equation and stored for future usage.
   - The result is 'a8f17d4b41ab8dab6c95d3c14e34a9e1'.
5. Next, the par called A2 needs to be evaluated:
   - The method type is 'REGISTER'.
   - Using SIP protocol 'sip'.
   - Proxy IP from *ini* file is '10.2.2.222'.
   - The equation to be evaluated is **'REGISTER:sip:10.2.2.222'.**
   - The MD5 algorithm is run on this equation and stored for future usage.
   - The result is 'a9a031cfddcb10d91c8e7b4926086f7e'.

6. Final stage:

   - The A1 result: The nonce from the proxy response is '11432d6bce58ddf02e3b5e1c77c010d2'.
   - The A2 result: The equation to be evaluated is **'A1:11432d6bce58ddf02e3b5e1c77c010d2:A2'.**
   - The MD5 algorithm is run on this equation. The outcome of the calculation is the response needed by the device to register with the Proxy.
   - The response is 'b9c45d0234a5abf5ddf5c704029b38cf'.

   At this time, a new REGISTER request is issued with the following response:

```
REGISTER sip:10.2.2.222 SIP/2.0
Via: SIP/2.0/UDP 10.1.1.200
From: <sip: 122@10.1.1.200>;tag=1c23940
To: <sip: 122@10.1.1.200>
Call-ID: 654982194@10.1.1.200
Server: Audiocodes-Sip-Gateway/Mediant 1000/v.6.00.010.006
CSeq: 1 REGISTER
Contact: sip:122@10.1.1.200:
Expires:3600
Authorization: Digest, username: 122,
realm="audiocodes.com",
nonce="11432d6bce58ddf02e3b5e1c77c010d2",
uri="10.2.2.222",
response="b9c45d0234a5abf5ddf5c704029b38cf"
```

7. Upon receiving this request and if accepted by the Proxy, the proxy returns a 200 OK response closing the REGISTER transaction:

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.1.1.200
From: <sip: 122@10.1.1.200>;tag=1c23940
To: <sip: 122@10.1.1.200>
Call-ID: 654982194@10.1.1.200
Cseq: 1 REGISTER
Date: Thu, 26 Jul 2001 09:34:42 GMT
Server: Columbia-SIP-Server/1.17
Content-Length: 0
Contact: <sip:122@10.1.1.200>; expires="Thu, 26 Jul 2001 10:34:42
GMT"; action=proxy; q=1.00
Contact: <122@10.1.1.200:>; expires="Tue, 19 Jan 2038 03:14:07
GMT"; action=proxy; q=0.00
Expires: Thu, 26 Jul 2001 10:34:42 GMT
```

9.11.3 Proxy or Registrar Registration Example

Below is an example of Proxy and Registrar registration:

```
REGISTER sip:servername SIP/2.0
VIA: SIP/2.0/UDP 212.179.22.229;branch=z9hG4bRaC7AU234
From: <sip:GWRegistrationName@sipgatewayname>;tag=1c29347
To: <sip:GWRegistrationName@sipgatewayname>
Call-ID: 10453@212.179.22.229
Seq: 1 REGISTER
Expires: 3600
Contact: sip:GWRegistrationName@212.179.22.229
Content-Length: 0
```

The ‘servername’ string is defined according to the following rules:

- The "servername" is equal to "RegistrarName" if configured. The "RegistrarName" can be any string.
- Otherwise, the "servername" is equal to "RegistrarIP" (either FQDN or numerical IP address), if configured.
- Otherwise, the "servername" is equal to "ProxyName" if configured. The "ProxyName" can be any string.
- Otherwise, the "servername" is equal to "ProxyIP" (either FQDN or numerical IP address).

The parameter GWRegistrationName can be any string. This parameter is used only if registration is per device. If the parameter is not defined, the parameter UserName is used instead. If the registration is per endpoint, the endpoint phone number is used.

The 'sipgatewayname' parameter (defined in the *ini* file or Web interface) can be any string. Some Proxy servers require that the 'sipgatewayname' (in REGISTER messages) is set equal to the Registrar / Proxy IP address or to the Registrar / Proxy domain name. The 'sipgatewayname' parameter can be overwritten by the TrunkGroupSettings_GatewayName value if the TrunkGroupSettings_RegistrationMode is set to 'Per Endpoint'.

REGISTER messages are sent to the Registrar's IP address (if configured) or to the Proxy's IP address. A single message is sent once per device, or messages are sent per B-channel according to the parameter AuthenticationMode. There is also an option to configure registration mode per Trunk Group using the TrunkGroupSettings table. The registration request is resent according to the parameter RegistrationTimeDivider. For example, if RegistrationTimeDivider = 70 (%) and Registration Expires time = 3600, the device resends its registration request after 3600 x 70% = 2520 sec. The default value of RegistrationTimeDivider is 50%.

If registration per B-channel is selected, on device startup the device sends REGISTER requests according to the maximum number of allowed SIP dialogs (configured by the parameter NumberOfActiveDialogs). After each received response, the subsequent REGISTER request is sent.

9.11.4 Establishing a Call between Two Devices

This section provides an example on configuring two AudioCodes' devices with FXS interfaces for establishing call communication. After configuration, you can make calls between telephones connected to the same device and between the two devices.

This example assumes the following:

- The IP address of the first device is 10.2.37.10 and its endpoint numbers are 101 to 104.
- The IP address of the second device is 10.2.37.20 and its endpoint numbers are 201 to 204.
- A SIP Proxy is not used. Internal call routing is performed using the device's 'Outbound IP Routing Table'.

➢ **To configure the two devices for call communication:**

1. For the *first* device (10.2.37.10), in the 'Trunk Group Table' page (refer to "Configuring the Trunk Group Table" on page 102), assign the phone numbers 101 to 104 to the device's endpoints.

Figure 9-57: Assigning Phone Numbers to Device 10.2.37.10

Group Index	Module	From Trunk	To Trunk	Channels	Phone Number	Trunk Group ID
1	Module 3 FXS	1	1	1-4	101	0

2. For the *second* device (10.2.37.20), in the 'Trunk Group Table' page, assign the phone numbers 201 to 204 to the device's endpoints.

Figure 9-58: Assigning Phone Numbers to Device 10.2.37.20

Group Index	Module	From Trunk	To Trunk	Channels	Phone Number	Trunk Group ID
1	Module 3 FXS	1	1	1-4	201	0

3. Configure the following settings for *both* devices:

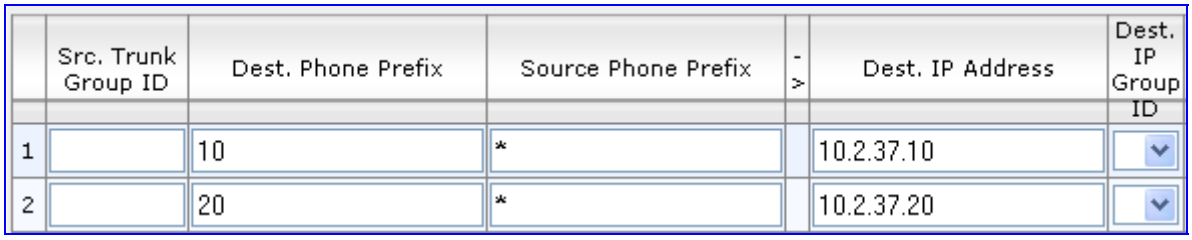
   In the 'Outbound IP Routing Table' page (refer to "Configuring the Outbound IP Routing Table" on page 158), add the following routing rules:

   a. In the first row, enter 10 for the destination phone prefix and enter 10.2.37.10 for the destination IP address (i.e., IP address of the first device).

   b. In the second row, enter 20 for the destination phone prefix and 10.2.37.20 for the destination IP address (i.e., IP address of the second device).

   These settings enable the routing (from both devices) of outgoing Tel-to-IP calls that start with 10 to the first device and calls that start with 20 to the second device.

Figure 9-59: Routing Calls Between Devices



	Src. Trunk Group ID	Dest. Phone Prefix	Source Phone Prefix	->	Dest. IP Address	Dest. IP Group ID
1		10	*		10.2.37.10	
2		20	*		10.2.37.20	

4. Make a call. Pick up the phone connected to port #1 of the first device and dial 102 (to the phone connected to port #2 of the same device). Listen for progress tones at the calling phone and for the ringing tone at the called phone. Answer the called phone, speak into the calling phone, and check the voice quality. Dial 201 from the phone connected to port #1 of the first device; the phone connected to port #1 of the second device rings. Answer the call and check the voice quality.

9.11.5 Trunk-to-Trunk Routing Example

This example describes two devices, each interfacing with the PSTN through four E1 spans. Device **A** is configured to route all incoming Tel-to-IP calls to Device **B**. Device **B** generates calls to the PSTN on the same E1 trunk on which the call was originally received (in Device **A**).

- Device **A** IP address: 192.168.3.50
- Device **B** IP address: 192.168.3.51

The *ini* file parameters configuration for devices **A** and **B** are as follows:

1. At both devices, define four Trunk Groups, each with 30 B-channels:
   - TrunkGroup_1 = 0/1-31,1000
   - TrunkGroup_2 = 1/1-31,2000
   - TrunkGroup_3 = 2/1-31,3000
   - TrunkGroup_4 = 3/1-31,4000
2. At Device **A**, add the originating Trunk Group ID as a prefix to the destination number for Tel-to-IP calls:

   AddTrunkGroupAsPrefix = 1
3. At Device **A**, route all incoming PSTN calls starting with prefixes 1, 2, 3, and 4, to the IP address of Device **B**:
   - Prefix = 1, 192.168.3.51
   - Prefix = 2, 192.168.3.51
   - Prefix = 3, 192.168.3.51
   - Prefix = 4, 192.168.3.51

   Note: You can also define Prefix = *,192.168.3.51, instead of the four lines above.
4. At Device **B**, route IP-to-PSTN calls to Trunk Group ID according to the first digit of the called number:
   - PSTNPrefix = 1,1
   - PSTNPrefix = 2,2

- PSTNPrefix = 3,4
- PSTNPrefix = 4,4

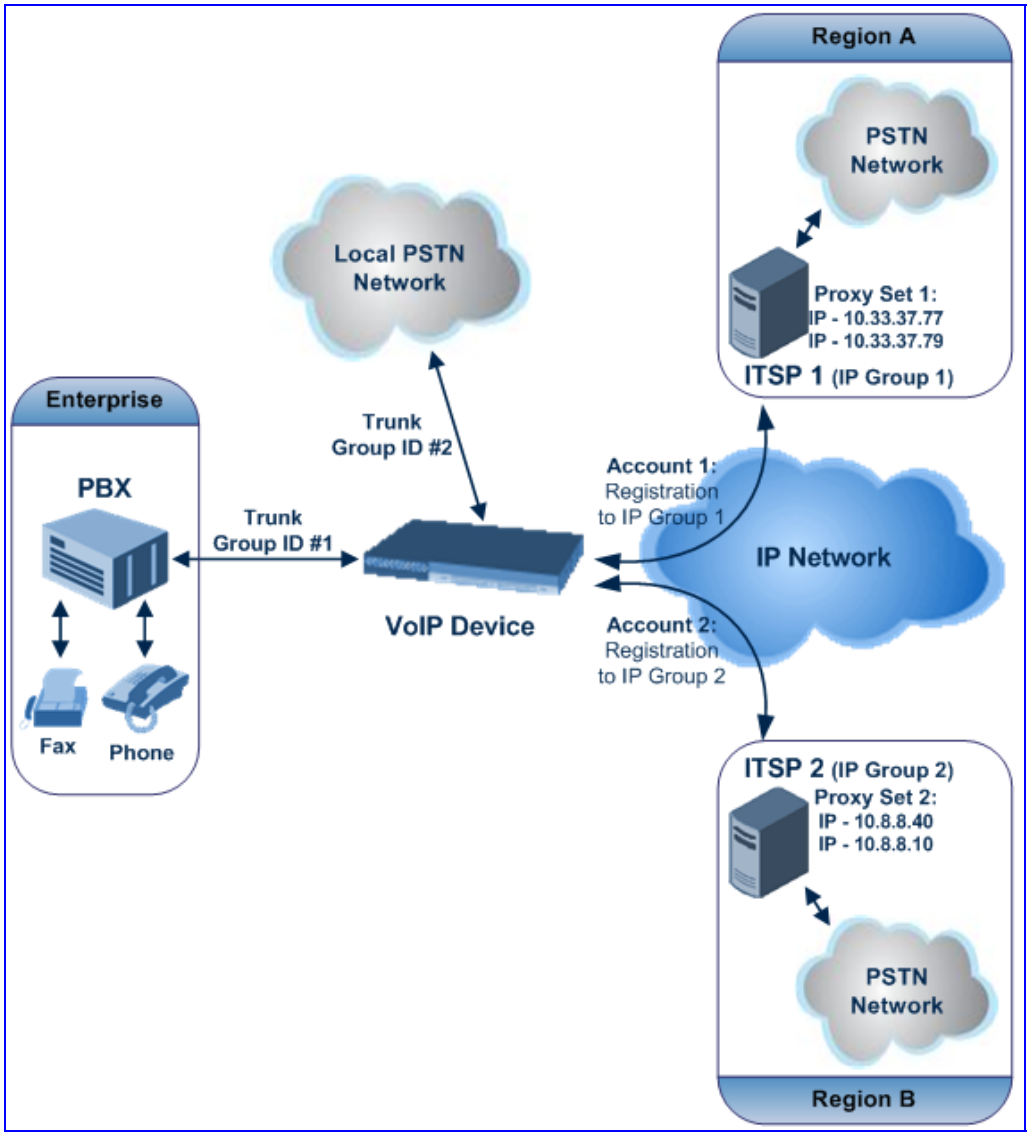
5. At Device **B**, remove the first digit from each IP-to-PSTN number before it is used in an outgoing call: NumberMapIP2Tel = *,1.

9.11.6 SIP Trunking between Enterprise and ITSPs

By implementing the device's enhanced and flexible routing capabilities, you can "design" complex routing schemes. This section provides an example of an elaborate routing scheme for SIP trunking between an Enterprise's PBX and two Internet Telephony Service Providers (ITSP), using AudioCodes' device.

Scenario: In this example, the Enterprise wishes to connect its TDM PBX to two different ITSPs, by implementing the device in its network environment. It's main objective is for the device to route Tel-to-IP calls to these ITSPs according to a dial plan. The device is to register (on behalf of the PBX) to each ITSP, which implements two servers for redundancy and load balancing. The Register messages must use different URI's in the From, To, and Contact headers per ITSP. In addition, all calls dialed from the Enterprise PBX with prefix '02' is sent to the local PSTN. The figure below illustrates this example setup:

Figure 9-60: Configuring Proxy Set ID #1 in the Proxy Sets Table Page

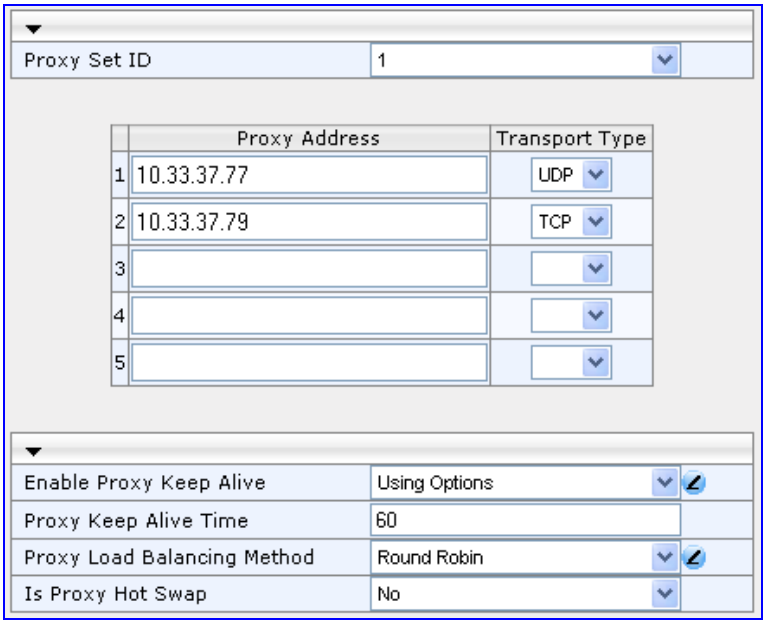


➢ **To configure call routing between an Enterprise and two ITSPs:**

1. Enable the device to register to a Proxy/Registrar server using the parameter IsRegisterNeeded.
2. In the 'Proxy Sets Table' page (refer to "Configuring the Proxy Sets Table" on page 123), configure two Proxy Sets and for each, enable Proxy Keep-Alive (using SIP OPTIONS) and 'round robin' load-balancing method:
   - Proxy Set **#1** includes two IP addresses of the first ITSP (**ITSP 1**) - 10.33.37.77 and 10.33.37.79 - and using UDP.
   - Proxy Set **#2** includes two IP addresses of the second ITSP (**ITSP 2**) - 10.8.8.40 and 10.8.8.10 - and using TCP.

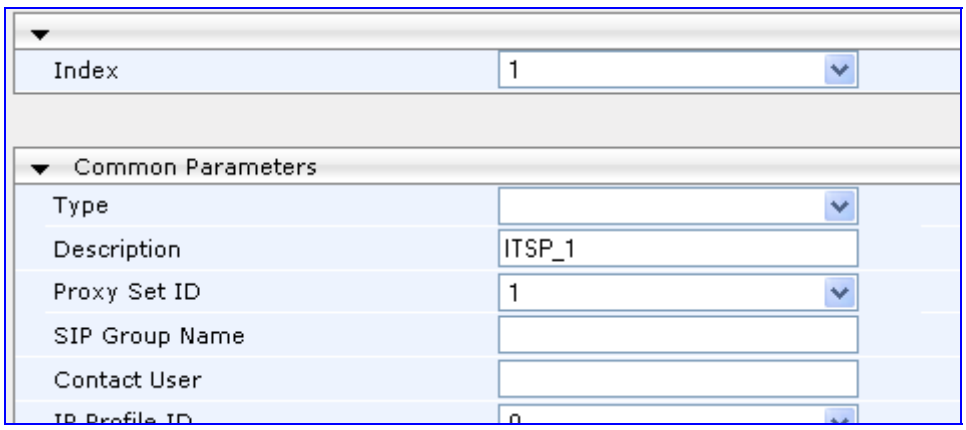
   The figure below displays the configuration of Proxy Set ID #1. Perform similar configuration for Proxy Set ID #2, but using different IP addresses.

Figure 9-61: Configuring Proxy Set ID #1 in the Proxy Sets Table Page



3. In the 'IP Group Table' page (refer to "Configuring the IP Groups" on page 113), configure the two IP Groups #1 and #2. Assign Proxy Sets #1 and #2 to IP Groups #1 and #2 respectively.

Figure 9-62: Configuring IP Groups #1 and #2 in the IP Group Table Page



4. In the 'Trunk Group Table' page, enable the Trunks connected between the Enterprise's PBX and the device (Trunk Group ID #1), and between the local PSTN and the device (Trunk Group ID #2).

Figure 9-63: Assigning Trunks to Trunk Group ID #1

Group Index	Module	From Trunk	To Trunk	Channels	Phone Number	Trunk Group ID	Tel Profile ID
1	Module 1 PRI	1	1	1-30	1100	1	0
2	Module 1 PRI	2	2	1-30	2200	2	0

5. In the 'Trunk Group Settings' page, configure 'Per Account' registration for Trunk Group ID #1 (without serving IP Group)

Figure 9-64: Configuring Trunk Group #1 for Registration per Account in Trunk Group Settings Page

	Trunk Group ID	Channel Select Mode	Registration Mode	Serving IP Group ID	Gateway Name	Contact User
1	1	Cyclic Ascending	Per Account			username

6. In the 'Account Table' page, configure the two Accounts for PBX trunk registration to ITSPs using the same Trunk Group (i.e., ID #1), but different serving IP Groups #1 and #2. For each account, define user name, password, and hostname, and ContactUser. The Register messages use different URI's (Hostname and ContactUser) in the From, To, and Contact headers per ITSP. Enable registration for both accounts.

Figure 9-65: Configuring Accounts for PBX Registration to ITSPs in Account Table Page

Index	ServedTrunkGroup	ServingIPGroup	Username	Password	HostName	Register	ContactUser
1	1	1	user1	1234	ITSP1	1	ITSP1user
2	1	2	user2	5555	ITSP2	1	ITSP2user

7. In the 'Inbound IP Routing Table' page, configure IP-to-Tel routing for calls from ITSPs to Trunk Group ID #1 (see 1 below) and from the device to the local PSTN (see 2 below).

Figure 9-66: Configuring ITSP-to-Trunk Group #1 Routing in IP to Trunk Group Table Page

	Dest. Host Prefix	Source Host Prefix	Dest. Phone Prefix	Source Phone Prefix	Source IP Address	->	Trunk Group ID	IP Profile ID	Source IPGroup ID
1			*	*			1		
2			02	*			2		

8. In the 'Outbound IP Routing Table' page, configure Tel-to-IP routing rules for calls to ITSPs (see first entry below) and to local PSTN (see second and third entries below).

Figure 9-67: Configuring Tel-to-IP Routing to ITSPs in Tel to IP Routing Table Page

Src. Trunk Group ID	Dest. Phone Prefix	Source Phone Prefix	->	Dest. IP Address	Port	Transport Type	Dest. IPGroup ID
1	0[3,4,5[	*				Not Configured	1
1	0[6,7,8]	*				Not Configured	2
1	02	*		10.13.4.13		Not Configured	

9.12 Mapping PSTN Release Cause to SIP Response

The device's FXO interface interoperates between the SIP network and the PSTN/PBX. This interoperability includes the mapping of PSTN/PBX Call Progress Tones to SIP 4xx or 5xx responses for IP-to-Tel calls. The converse is also true - for Tel-to-IP calls, the SIP 4xx or 5xx responses are mapped to tones played to the PSTN/PBX.

When establishing an IP-to-Tel call, the following rules are applied:

- If the remote party (PSTN/PBX) is busy and the FXO device detects a Busy tone, it sends a SIP 486 Busy response to IP. If it detects a Reorder tone, it sends a SIP 404 Not Found (no route to destination) to IP. In both cases, the call is released. Note that if the parameter DisconnectOnBusyTone is set to 0, the FXO device ignores the detection of Busy/Reorder tones and doesn't release the call.
- For all other FXS/FXO release types (caused when there are no free channels in the specific Trunk Group), or when an appropriate rule for routing the call to a Trunk Group doesn't exist, or if the phone number isn't found), the device sends a SIP response (to IP) according to the parameter DefaultReleaseCause. This parameter defines Q.931 release causes. Its default value is '3', which is mapped to the SIP 404 response. By changing its value to '34', the SIP 503 response is sent. Other causes can be used as well.

9.13 Querying Device Channel Resources using SIP OPTIONS

The device reports its maximum and available channel resources in SIP 200 OK responses upon receipt of SIP OPTIONS messages. The device sends this information in the SIP X-Resources header with the following parameters:

- **telchs:** specifies the total telephone channels as well as the number of free (available) telephone channels
- **mediachs:** not applicablespecifies the total and the free number of channels associated with media services (e.g., announcements and conferencing)

Below is an example of the X-Resources:

```
X-Resources: telchs= 12/4;mediachs=0/0
```

In the example above, "telchs" specifies the number of available channels and the number of occupied channels (4 channels are occupied and 12 channels are available).

9.14 Answer Machine Detector (AMD)

Answering Machine Detection (AMD) can be useful in automatic dialing applications. In some of these applications, it is important to detect if a human voice or an answering machine is answering the call. AMD can be activated and de-activated only after a channel is already open. The direction of the detection (PSTN or IP) can be configured using the parameter AMDDetectionDirection.

The sensitivity level of detection (voice and/or fax) can be configured between 0 and 7 for "Normal" sensitivity levels (using the parameter AMDDetectionSensitivity), or between 0 and 15 for "High" sensitivity (using the parameter AMDDetectionSensitivityHighResolution). The type of sensitivity ("Normal" or "High") is configured using the parameter AMDSensitivityResolution.

The device also supports the detection of beeps at the end of an answering machine message. This allows users of certain third-party, Application servers to leave a voice message after an answering machine plays a “beep”.

The device supports two methods for detecting and reporting beeps (configured using the AMDBeepDetectionMode parameter):

- Using the AMD detector. This detector is integrated in the existing AMD feature. The beep detection timeout and beep detection sensitivity are configurable using the AMDBeepDetectionTimeout and AMDBeepDetectionSensitivity parameters respectively.
- Using the Call Progress Tone detector - several beep tones (Tone Type #46) can be configured in the CPT file.

The detection of beeps is done using the X-Detect header extension. The device sends a SIP INFO message containing one of the following field values:

- Type=AMD and SubType=Beep
- Type=CPT and SubType=Beep

Upon every AMD activation, the device can send a SIP INFO message to an Application server notifying it of one of the following:

- Human voice has been detected
- Answering machine has been detected
- Silence (i.e., no voice detected) has been detected

The table below shows the success rates of the AMD feature for correctly detecting live and fax calls:

Table 9-5: Approximate AMD Detection Normal Sensitivity (Based on North American English)

AMD Detection Sensitivity	Performance	
	Success Rate for Live Calls	Success Rate for Answering Machine
0 (Best for Answering Machine)	-	-
1	82.56%	97.10%
2	85.87%	96.43%
3 (Default)	88.57%	94.76%

AMD Detection Sensitivity	Performance	
	Success Rate for Live Calls	Success Rate for Answering Machine
4	88.94%	94.31%
5	90.42%	91.64%
6	90.66%	91.30%
7 (Best for Live Calls)	94.72%	76.14%

Table 9-6: Approximate AMD Detection High Sensitivity (Based on North American English)

AMD Detection Sensitivity	Performance	
	Success Rate for Live Calls	Success Rate for Answering Machine
0 (Best for Answering Machine)	72%	97%
1	77%	96%
2	79%	95%
3	80%	95%
4	84%	94%
5	86%	93%
6	87%	92%
7	88%	91%
8 (default)	90%	89%
9	90%	88%
10	91%	87%
11	94%	78%
12	94%	73%
13	95%	65%
14	96%	62%
15 (Best for Live Calls)	97%	46%

A pre-requisite for enabling the AMD feature is to set the *ini* file parameter EnableDSPIPMDetectors to 1. In addition, to enable voice detection, required once the AMD detects the answering machine, the *ini* file parameter EnableVoiceDetection must be set to 1.

Note: The device's AMD feature is based on voice detection for North American English. If you want to implement AMD in a different language or region, you must provide AudioCodes with a database of recorded voices in the language on which the device's AMD mechanism can base its voice detector algorithms for detecting these voices. The data needed for an accurate calibration should be recorded under the following guidelines:

- Statistical accuracy: The number of recordings should be large (i.e., about 100) and varied. The calls must be made to different people, at different times. The calls must be made in the specific location in which the device's AMD mechanism is to operate.
- Real-life recording: The recordings should simulate real-life answering of a person picking up the phone without the caller speaking (until the AMD decision).
- Normal environment interferences: The environment should almost simulate real-life scenarios, i.e., not sterile but not too noisy either. Interferences, for example, could include background noises of other people talking, spikes, and car noises.

The SIP call flows below show an example of implementing the device's AMD feature. This scenario example allows a third-party Application server to play a recorded voice message to an answering machine.

1. Upon detection by the device of the answering machine, the device sends a SIP INFO message to the Application server:

```
INFO sip:sipp@172.22.2.9:5060 SIP/2.0
Via: SIP/2.0/UDP 172.22.168.249;branch=z9hG4bKac1566945480
Max-Forwards: 70
From: sut <sip:3000@172.22.168.249:5060>;tag=1c1505895240
To: sipp <sip:sipp@172.22.2.9:5060>;tag=1
Call-ID: 1-29758@172.22.2.9
CSeq: 1 INFO
Contact: <sip:56700@172.22.168.249>
Supported: em,timer,replaces,path,resource-priority
Allow:
REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,INFO,SUB
SCRIBE,UPDATE
User-Agent: Audiocodes-Sip-Gateway/v.6.00A.040.004
Content-Type: application/x-detect
Content-Length: 30
Type= AMD
SubType= AUTOMATA
```

2. The device then detects the start of voice (i.e., the greeting message of the answering machine), and then sends the following to the Application server:

```
INFO sip:sipp@172.22.2.9:5060 SIP/2.0
Via: SIP/2.0/UDP 172.22.168.249;branch=z9hG4bKac482466515
Max-Forwards: 70
From: sut <sip:3000@172.22.168.249:5060>;tag=1c419779142
To: sipp <sip:sipp@172.22.2.9:5060>;tag=1
Call-ID: 1-29753@172.22.2.9
CSeq: 1 INFO
Contact: <sip:56700@172.22.168.249>
Supported: em,timer,replaces,path,resource-priority
Allow:
REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,INFO,SUB
SCRIBE,UPDATE
User-Agent: Audiocodes-Sip-Gateway/v.6.00A.040.004
Content-Type: application/x-detect
Content-Length: 34
Type= PTT
SubType= SPEECH-START
```

3. Upon detection of the end of voice (i.e., end of the greeting message of the answering machine), the device sends the Application server the following:

```
INFO sip:sipp@172.22.2.9:5060 SIP/2.0
Via: SIP/2.0/UDP 172.22.168.249;branch=z9hG4bKac482466515
Max-Forwards: 70
From: sut <sip:3000@172.22.168.249:5060>;tag=1c419779142
To: sipp <sip:sipp@172.22.2.9:5060>;tag=1
Call-ID: 1-29753@172.22.2.9
CSeq: 1 INFO
Contact: <sip:56700@172.22.168.249>
Supported: em,timer,replaces,path,resource-priority
Allow:
REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,INFO,SUB
SCRIBE,UPDATE
User-Agent: Audiocodes-Sip-Gateway/v.6.00A.040.004
Content-Type: application/x-detect
Content-Length: 34
Type= PTT
SubType= SPEECH-END
```

4. The Application server now sends its message to the answering message.

If the device detects voice and not an answering machine, the SIP INFO message includes:

```
Type= AMD
SubType= VOICE
```

If the device detects silence, the SIP INFO message includes the SubType **SILENT**.

9.15 Event Notification using X-Detect Header

The device supports the sending of notifications to a remote party notifying the occurrence (or detection) of certain events on the media stream. Event detection and notifications is performed using the SIP X-Detect message header and only when establishing a SIP dialog.

For supporting some events, certain device configurations need to be performed. The table below lists the supported event types (and subtypes) and the corresponding device configurations, if required:

Table 9-7: Supported X-Detect Event Types

Events Type	Subtype	Required Configuration
AMD	voice automatic silence unknown beep	EnableDSPIPMDetectors = 1 AMDTimeout = 2000 (msec) For AMD beep detection, AMDBeepDetectionMode = 1 or 2
CPT	SIT-NC SIT-IC SIT-VC SIT-RO Busy Reorder Ringtone beep	SITDetectorEnable = 1 UserDefinedToneDetectorEnable = 1 **Notes:** ▪ Ensure that the CPT file is configured with the required tone type. ▪ On beep detection, a SIP INFO message is sent with type AMD/CPT and subtype beep. ▪ The beep detection must be started using regular X-detect extension, with AMD or CPT request.
FAX	CED	(IsFaxUsed ≠ 0) or (IsFaxUsed = 0, and FaxTransportMode ≠ 0)
	modem	VxxModemTransportType = 3
PTT	voice-start voice-end	EnableDSPIPMDetectors = 1

The device can detect and report the following Special Information Tones (SIT) types from the PSTN:

- SIT-NC (No Circuit found)
- SIT-IC (Operator Intercept)
- SIT-VC (Vacant Circuit - non-registered number)
- SIT-RO (Reorder - System Busy)

There are additional three SIT tones that are detected as one of the above SIT tones:

- The NC* SIT tone is detected as NC
- The RO* SIT tone is detected as RO
- The IO* SIT tone is detected as VC

The device can map these SIT tones to a Q.850 cause and then map them to SIP 5xx/4xx responses, using the parameters SITQ850Cause, SITQ850CauseForNC, SITQ850CauseForIC, SITQ850CauseForVC, and SITQ850CauseForRO.

Table 9-8: Special Information Tones (SITs) Reported by the device

Special Information Tones (SITs) Name	Description	First Tone Frequency Duration		Second Tone Frequency Duration		Third Tone Frequency Duration	
		(Hz)	(ms)	(Hz)	(ms)	(Hz)	(ms)
NC1	No circuit found	985.2	380	1428.5	380	1776.7	380
IC	Operator intercept	913.8	274	1370.6	274	1776.7	380
VC	Vacant circuit (non registered number)	985.2	380	1370.6	274	1776.7	380
RO1	Reorder (system busy)	913.8	274	1428.5	380	1776.7	380
NC*	-	913.8	380	1370.6	380	1776.7	380
RO*	-	985.2	274	1370.6	380	1776.7	380
IO*	-	913.8	380	1428.5	274	1776.7	380

For example:

```
INFO sip:5001@10.33.2.36 SIP/2.0
Via: SIP/2.0/UDP 10.33.45.65;branch=z9hG4bKac2042168670
Max-Forwards: 70
From: <sip:5000@10.33.45.65;user=phone>;tag=1c1915542705
To: <sip:5001@10.33.2.36;user=phone>;tag=WQJNIDDPCOKAPIDSCOTG
Call-ID: AIFHPETLLMVVFWPDXUHD@10.33.2.36
CSeq: 1 INFO
Contact: <sip:2206@10.33.45.65>
Supported: em,timer,replaces,path,resource-priority
Content-Type: application/x-detect
Content-Length: 28
Type= CPT
SubType= SIT-IC
```

The X-Detect event notification process is as follows:

1. For IP-to-Tel or Tel-to-IP calls, the device receives a SIP request message (using the X-Detect header) that the remote party wishes to detect events on the media stream. For incoming (IP-to-Tel) calls, the request must be indicated in the initial INVITE and responded to either in the 183 response (for early dialogs) or in the 200 OK response (for confirmed dialogs). For outgoing calls (Tel-to-IP), the request may be received in the 183 (for early dialogs) and responded to in the PRACK, or received in the 200 OK (for confirmed dialogs) and responded to in the ACK.
2. Once the device receives such a request, it sends a SIP response message (using the X-Detect header) to the remote party, listing all supported events that can be detected. The absence of the X-Detect header indicates that no detections are available.

3. Each time the device detects a supported event, the event is notified to the remote party by sending an INFO message with the following message body:
   - Content-Type: application/X-DETECT
   - Type = **[AMD |** CPT | FAX | PTT…]
   - Subtype = xxx (according to the defined subtypes of each type)

Below is an example of SIP messages using the X-Detect header:

```
INVITE sip:101@10.33.2.53;user=phone SIP/2.0
Via: SIP/2.0/UDP 10.33.2.53;branch=z9hG4bKac5906
Max-Forwards: 70
From: "anonymous" <sip:anonymous@anonymous.invalid>;tag=1c25298
To: <sip:101@10.33.2.53;user=phone>
Call-ID: 11923@10.33.2.53
CSeq: 1 INVITE
Contact: <sip:100@10.33.2.53>
X- Detect: Request=CPT,FAX
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.33.2.53;branch=z9hG4bKac5906
From: "anonymous" <sip:anonymous@anonymous.invalid>;tag=1c25298
To: <sip:101@10.33.2.53;user=phone>;tag=1c19282
Call-ID: 11923@10.33.2.53
CSeq: 1 INVITE
Contact: <sip:101@10.33.2.53>
X- Detect: Response=CPT,FAX
INFO sip:101@10.33.2.53;user=phone SIP/2.0
Via: SIP/2.0/UDP 10.33.2.53;branch=z9hG4bKac5906
Max-Forwards: 70
From: "anonymous" <sip:anonymous@anonymous.invalid>;tag=1c25298
To: <sip:101@10.33.2.53;user=phone>
Call-ID: 11923@10.33.2.53
CSeq: 1 INVITE
Contact: <sip:100@10.33.2.53>
X- Detect: Response=CPT,FAX
Content-Type: Application/X-Detect
Content-Length: xxx
Type = CPT
Subtype = SIT
```

9.16 Supported RADIUS Attributes

The following table provides explanations on the RADIUS attributes included in the communication packets transmitted between the device and a RADIUS Server.

Table 9-9: Supported RADIUS Attributes

Attribute Number	Attribute Name	VSA No.	Purpose	Value Format	Example	AAA[1]
Request Attributes						
1	User-Name		Account number or calling party number or blank	String up to 15 digits long	5421385747	Start Acc Stop Acc
4	NAS-IP-Address		IP address of the requesting device	Numeric	192.168.14.43	Start Acc Stop Acc

Attribute Number	Attribute Name	VSA No.	Purpose	Value Format	Example	AAA[1]
6	Service-Type		Type of service requested	Numeric	1: login	Start Acc Stop Acc
26	H323-Incoming-Conf-Id	1	SIP call identifier	Up to 32 octets		Start Acc Stop Acc
26	H323-Remote-Address	23	IP address of the remote gateway	Numeric		Stop Acc
26	H323-Conf-ID	24	H.323/SIP call identifier	Up to 32 octets		Start Acc Stop Acc
26	H323-Setup-Time	25	Setup time in NTP format 1	String		Start Acc Stop Acc
26	H323-Call-Origin	26	The call's originator: Answering (IP) or Originator (PSTN)	String	Answer, Originate etc	Start Acc Stop Acc
26	H323-Call-Type	27	Protocol type or family used on this leg of the call	String	VoIP	Start Acc Stop Acc
26	H323-Connect-Time	28	Connect time in NTP format	String		Stop Acc
26	H323-Disconnect-Time	29	Disconnect time in NTP format	String		Stop Acc
26	H323-Disconnect-Cause	30	Q.931 disconnect cause code	Numeric		Stop Acc
26	H323-Gw-ID	33	Name of the gateway	String	SIPIDString	Start Acc Stop Acc
26	SIP-Call-ID	34	SIP Call ID	String	abcde@ac.com	Start Acc Stop Acc
26	Call-Terminator	35	The call's terminator: PSTN-terminated call (Yes); IP-terminated call (No).	String	Yes, No	Stop Acc

Attribute Number	Attribute Name	VSA No.	Purpose	Value Format	Example	AAA[1]
30	Called-Station-ID			String	8004567145	Start Acc
			Destination phone number	String	2427456425	Stop Acc
			Calling Party Number (ANI)	String	5135672127	Start Acc Stop Acc
			Account Request Type (start or stop) **Note:** 'start' isn't supported on the Calling Card application.	Numeric	1: start, 2: stop	Start Acc Stop Acc
			No. of seconds tried in sending a particular record	Numeric	5	Start Acc Stop Acc
			Number of octets received for that call duration	Numeric		Stop Acc
			Number of octets sent for that call duration	Numeric		Stop Acc
			A unique accounting identifier - match start & stop	String	34832	Start Acc Stop Acc
			For how many seconds the user received the service	Numeric		Stop Acc
			Number of packets received during the call	Numeric		Stop Acc
			Number of packets sent during the call	Numeric		Stop Acc
			Physical port type of device on which the call is active	String	0: Asynchronous	Start Acc Stop Acc
Response Attributes						
26	H323-Return-Code	103	The reason for failing authentication (0 = ok, other number failed)	Numeric	0 Request accepted	Stop Acc
44	Acct-Session-ID		A unique accounting identifier – match start & stop	String		Stop Acc

Below is an example of RADIUS Accounting, where the non-standard parameters are preceded with brackets.

```
Accounting-Request (361)
user-name = 111
acct-session-id = 1
nas-ip-address = 212.179.22.213
nas-port-type = 0
acct-status-type = 2
acct-input-octets = 4841
acct-output-octets = 8800
acct-session-time = 1
acct-input-packets = 122
acct-output-packets = 220
called-station-id = 201
calling-station-id = 202
// Accounting non-standard parameters:
(4923 33) h323-gw-id =
(4923 23) h323-remote-address = 212.179.22.214
(4923 1) h323-ivr-out = h323-incoming-conf-id:02102944 600a1899
3fd61009 0e2f3cc5
(4923 30) h323-disconnect-cause = 22 (0x16)
(4923 27) h323-call-type = VOIP
(4923 26) h323-call-origin = Originate
(4923 24) h323-conf-id = 02102944 600a1899 3fd61009 0e2f3cc5
```

9.17 Call Detail Record

The Call Detail Record (CDR) contains vital statistic information on calls made by the device. CDRs are generated at the end and (optionally) at the beginning of each call (determined by the parameter CDRReportLevel), and then sent to a Syslog server. The destination IP address for CDR logs is determined by the parameter CDRSyslogServerIP. For CDR in RADIUS format, refer to "Supported RADIUS Attributes" on page 596.

The following table lists the supported CDR fields.

Table 9-10: Supported CDR Fields

Field Name	Description
ReportType	Report for either Call Started, Call Connected, or Call Released
Cid	Port Number
CallId	SIP Call Identifier
Trunk	Physical Trunk Number
BChan	Selected B-Channel
ConId	SIP Conference ID
TG	Trunk Group Number
EPTyp	Endpoint Type
Orig	Call Originator (IP, Tel)
SourceIp	Source IP Address
DestIp	Destination IP Address

Field Name	Description
TON	Source Phone Number Type
NPI	Source Phone Number Plan
SrcPhoneNum	Source Phone Number
SrcNumBeforeMap	Source Number Before Manipulation
TON	Destination Phone Number Type
NPI	Destination Phone Number Plan
DstPhoneNum	Destination Phone Number
DstNumBeforeMap	Destination Number Before Manipulation
Durat	Call Duration
Coder	Selected Coder
Intrv	Packet Interval
RtpIp	RTP IP Address
Port	Remote RTP Port
TrmSd	Initiator of Call Release (IP, Tel, Unknown)
TrmReason	Termination Reason
Fax	Fax Transaction during the Call
InPackets	Number of Incoming Packets
OutPackets	Number of Outgoing Packets
PackLoss	Local Packet Loss
RemotePackLoss	Number of Outgoing Lost Packets
UniqueId	unique RTP ID
SetupTime	Call Setup Time
ConnectTime	Call Connect Time
ReleaseTime	Call Release Time
RTPdelay	RTP Delay
RTPjitter	RTP Jitter
RTPssrc	Local RTP SSRC
RemoteRTPssrc	Remote RTP SSRC
RedirectReason	Redirect Reason
TON	Redirection Phone Number Type
MeteringPulses	Number of Generated Metering Pulses
NPI	Redirection Phone Number Plan
RedirectPhonNum	Redirection Phone Number

9.18 RTP Multiplexing (ThroughPacket)

The device supports a proprietary method to aggregate RTP streams from several channels. This reduces the bandwidth overhead caused by the attached Ethernet, IP, UDP, and RTP headers and reduces the packet/data transmission rate. This option reduces the load on network routers and can typically save 50% (e.g., for G.723) on IP bandwidth. RTP Multiplexing (ThroughPacket™) is accomplished by aggregating payloads from several channels that are sent to the same destination IP address into a single IP packet.

RTP multiplexing can be applied to the entire device (refer to "Configuring the RTP/RTCP Settings" on page 74) or to specific IP destinations using the IP Profile feature (refer to "Configuring IP Profiles" on page 133).

To enable RTP Multiplexing, set the parameter RemoteBaseUDPPort to a non-zero value. Note that the value of RemoteBaseUDPPort on the local device must equal the value of BaseUDPPort of the remote device. The device uses these parameters to identify and distribute the payloads from the received multiplexed IP packet to the relevant channels.

In RTP Multiplexing mode, the device uses a single UDP port for all incoming multiplexed packets and a different port for outgoing packets. These ports are configured using the parameters L1L1ComplexTxUDPPort and L1L1ComplexRxUDPPort.

When RTP Multiplexing is used, call statistics are unavailable (since there is no RTCP flow).

Notes:

- RTP Multiplexing must be enabled on both devices.
- When VLANs are implemented, the RTP Multiplexing mechanism is not supported.

9.19 Dynamic Jitter Buffer Operation

Voice frames are transmitted at a fixed rate. If the frames arrive at the other end at the same rate, voice quality is perceived as good. In many cases, however, some frames can arrive slightly faster or slower than the other frames. This is called jitter (delay variation), and degrades the perceived voice quality. To minimize this problem, the device uses a jitter buffer. The jitter buffer collects voice packets, stores them and sends them to the voice processor in evenly spaced intervals.

The device uses a dynamic jitter buffer that can be configured using the following two parameters:

- **Minimum delay:** DJBufMinDelay (0 msec to 150 msec)
  Defines the starting jitter capacity of the buffer. For example, at 0 msec, there is no buffering at the start. At the default level of 10 msec, the device always buffers incoming packets by at least 10 msec worth of voice frames.
- **Optimization Factor:** DJBufOptFactor (0 to 12, 13)
  Defines how the jitter buffer tracks to changing network conditions. When set at its maximum value of 12, the dynamic buffer aggressively tracks changes in delay (based on packet loss statistics) to increase the size of the buffer and doesn't decay back down. This results in the best packet error performance, but at the cost of extra delay. At the minimum value of 0, the buffer tracks delays only to compensate for clock drift and quickly decays back to the minimum level. This optimizes the delay performance but at the expense of a higher error rate.

The default settings of 10 msec Minimum delay and 10 Optimization Factor should provide a good compromise between delay and error rate. The jitter buffer 'holds' incoming packets for 10 msec before making them available for decoding into voice. The coder polls frames from the buffer at regular intervals in order to produce continuous speech. As long as delays in the network do not change (jitter) by more than 10 msec from one packet to the next, there is always a sample in the buffer for the coder to use. If there is more than 10 msec of delay at any time during the call, the packet arrives too late. The coder tries to access a frame and is not able to find one. The coder must produce a voice sample even if a frame is not available. It therefore compensates for the missing packet by adding a Bad-Frame-Interpolation (BFI) packet. This loss is then flagged as the buffer being too small. The dynamic algorithm then causes the size of the buffer to increase for the next voice session. The size of the buffer may decrease again if the device notices that the buffer is not filling up as much as expected. At no time does the buffer decrease to less than the minimum size configured by the Minimum delay parameter.

For certain scenarios, the **Optimization Factor is set to 13**: One of the purposes of the Jitter Buffer mechanism is to compensate for clock drift. If the two sides of the VoIP call are not synchronized to the same clock source, one RTP source generates packets at a lower rate, causing under-runs at the remote Jitter Buffer. In normal operation (optimization factor 0 to 12), the Jitter Buffer mechanism detects and compensates for the clock drift by occasionally dropping a voice packet or by adding a BFI packet.

Fax and modem devices are sensitive to small packet losses or to added BFI packets. Therefore, to achieve better performance during modem and fax calls, the Optimization Factor should be set to 13. In this special mode the clock drift correction is performed less frequently - only when the Jitter Buffer is completely empty or completely full. When such condition occurs, the correction is performed by dropping several voice packets simultaneously or by adding several BFI packets simultaneously, so that the Jitter Buffer returns to its normal condition.

10 Networking Capabilities

This section provides an overview of the device's networking capabilities.

10.1 Ethernet Interface Configuration

The device's Ethernet connection can be configured (using the *ini* file parameter EthernetPhyConfiguration) for one of the following modes:

- **Manual mode:**
  - 10Base-T Half-Duplex or 10Base-T Full-Duplex
  - 100Base-TX Half-Duplex or 100Base-TX Full-Duplex
- **Auto-Negotiation:** chooses common transmission parameters such as speed and duplex mode

The Ethernet connection should be configured according to the following recommended guidelines:

- When the device's Ethernet port is configured for Auto-Negotiation, the opposite port must also operate in Auto-Negotiation. Auto-Negotiation falls back to Half-Duplex mode when the opposite port is not in Auto-Negotiation mode, but the speed (i.e., 10/100Base-T or 1000Base-TX) in this mode is always configured correctly. Configuring the device to Auto-Negotiation mode while the opposite port is set manually to Full-Duplex is invalid as it causes the device to fall back to Half-Duplex mode while the opposite port is Full-Duplex. Any mismatch configuration can yield unexpected functioning of the Ethernet connection.
- When configuring the device's Ethernet port manually, the same mode (i.e., Half Duplex or Full Duplex) and speed must be configured on the remote Ethernet port. In addition, when the device's Ethernet port is configured manually, it is invalid to set the remote port to Auto-Negotiation**.** Any mismatch configuration can yield unexpected functioning of the Ethernet connection.
- It's recommended to configure the port for best performance and highest bandwidth (i.e., Full Duplex with 100Base-TX), but at the same time adhering to the guidelines listed above.

Note that when remote configuration is performed, the device should be in the correct Ethernet setting prior to the time this parameter takes effect. When, for example, the device is configured using BootP/TFTP, the device performs many Ethernet-based transactions prior to reading the *ini* file containing this device configuration parameter. To resolve this problem, the device always uses the last Ethernet setup mode configured. In this way, if you want to configure the device to operate in a new network environment in which the current Ethernet setting of the device is invalid, you should first modify this parameter in the current network so that the new setting holds next time the device is restarted. After reconfiguration has completed, connect the device to the new network and restart it. As a result, the remote configuration process that occurs in the new network uses a valid Ethernet configuration.

10.2 Ethernet Interface Redundancy

The device supports Ethernet redundancy by providing two Ethernet ports, located on the CPU module. The Ethernet port redundancy feature is enabled using the *ini* file parameter MIIRedundancyEnable. By default, this feature is disabled.

When Ethernet redundancy is implemented, the two Ethernet ports can be connected to the same switch (segment / hub). In this setup, one Ethernet port is active and the other is redundant. If an Ethernet connection failure is detected, the CPU module switches over to the redundant Ethernet port. The CPU issues a Major alarm notifying of the failed physical port. If the first Ethernet port connection is restored, the Major alarm is cleared. The first physical port now becomes the redundant Ethernet port in case of failure with the active physical port (which is currently the second physical port).

When the CPU module loses all Ethernet connectivity, a Critical alarm is generated:

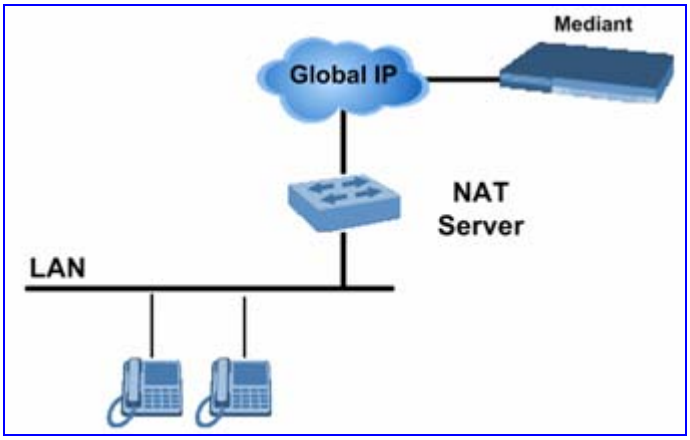
- When MIIRedundancyEnable is disabled: the alarm is generated when the single physical connection is lost. The alarm is cleared when the single physical connection is restored.
- When MIIRedundancyEnable is enabled: the alarm is generated when both physical connections are lost. The alarm is cleared when one or both of the physical connections are restored.

10.3 NAT (Network Address Translation) Support

Network Address Translation (NAT) is a mechanism that maps a set of internal IP addresses used within a private network to global IP addresses, providing transparent routing to end hosts. The primary advantages of NAT include (1) Reduction in the number of global IP addresses required in a private network (global IP addresses are only used to connect to the Internet); (2) Better network security by hiding its internal architecture.

The following figure illustrates the device's supported NAT architecture.

Figure 10-1: NAT Support



The design of SIP creates a problem for VoIP traffic to pass through NAT. SIP uses IP addresses and port numbers in its message body and the NAT server can't modify SIP messages and therefore, can't change local to global addresses. Two different streams traverse through NAT: signaling and media. A device (located behind a NAT) that initiates a signaling path has problems in receiving incoming signaling responses (they are blocked by the NAT server). Furthermore, the initiating device must notify the receiving device where to send the media.

To resolve these issues, the following mechanisms are available:

- STUN (refer to STUN on page 605)
- First Incoming Packet Mechanism (refer to "First Incoming Packet Mechanism" on page 606)
- RTP No-Op packets according to the avt-rtp-noop draft (refer to "No-Op Packets" on page 606)

For information on SNMP NAT traversal, refer to the *Product Reference Manual*.

10.3.1 STUN

Simple Traversal of UDP through NATs (STUN), based on RFC 3489 is a client / server protocol that solves most of the NAT traversal problems. The STUN server operates in the public Internet and the STUN clients are embedded in end-devices (located behind NAT). STUN is used both for the signaling and the media streams. STUN works with many existing NAT types and does not require any special behavior.

STUN enables the device to discover the presence (and types) of NATs and firewalls located between it and the public Internet. It provides the device with the capability to determine the public IP address and port allocated to it by the NAT. This information is later embedded in outgoing SIP / SDP messages and enables remote SIP user agents to reach the device. It also discovers the binding lifetime of the NAT (the refresh rate necessary to keep NAT 'Pinholes' open).

On startup, the device sends a STUN Binding Request. The information received in the STUN Binding Response (IP address:port) is used for SIP signaling. This information is updated every user-defined period (NATBindingDefaultTimeout).

At the beginning of each call and if STUN is required (i.e., not an internal NAT call), the media ports of the call are mapped. The call is delayed until the STUN Binding Response (that includes a global IP:port) for each media (RTP, RTCP and T.38) is received.

To enable STUN, perform the following:

- Enable the STUN feature(by setting the *ini* file parameter EnableSTUN to 1).
- Define the STUN server address using one of the following methods:
  - Define the IP address of the primary and the secondary (optional) STUN servers (using the *ini* file parameters STUNServerPrimaryIP and STUNServerSecondaryIP). If the primary STUN server isn't available, the device attempts to communicate with the secondary server.
  - Define the domain name of the STUN server using the *ini* file parameter StunServerDomainName. The STUN client retrieves all STUN servers with an SRV query to resolve this domain name to an IP address and port, sort the server list, and use the servers according to the sorted list.
- Use the *ini* file parameter NATBindingDefaultTimeout to define the default NAT binding lifetime in seconds. STUN is used to refresh the binding information after this time expires.

Notes:

- STUN only applies to UDP (it doesn't support TCP and TLS).
- STUN can't be used when the device is located behind a symmetric NAT.
- Use either the STUN server IP address (STUNServerPrimaryIP) or domain name (STUNServerDomainName) method, with priority to the first one.

10.3.2 First Incoming Packet Mechanism

If the remote device resides behind a NAT device, it's possible that the device can activate the RTP/RTCP/T.38 streams to an invalid IP address / UDP port. To avoid such cases, the device automatically compares the source address of the incoming RTP/RTCP/T.38 stream with the IP address and UDP port of the remote device. If the two are not identical, the transmitter modifies the sending address to correspond with the address of the incoming stream. The RTP, RTCP and T.38 can thus have independent destination IP addresses and UDP ports.

You can disable the NAT mechanism by setting the *ini* file parameter DisableNAT to 1. The two parameters EnableIpAddrTranslation and EnableUdpPortTranslation allow you to specify the type of compare operation that occurs on the first incoming packet. To compare only the IP address, set EnableIpAddrTranslation to 1, and EnableUdpPortTranslation to 0. In this case, if the first incoming packet arrives with only a difference in the UDP port, the sending addresses won't change. If both the IP address and UDP port need to be compared, then both parameters need to be set to 1.

10.3.3 No-Op Packets

The device's No-Op packet support can be used to verify Real-Time Transport Protocol (RTP) and T.38 connectivity, and to keep NAT bindings and Firewall pinholes open. The No-Op packets are available for sending in RTP and T.38 formats.

You can control the activation of No-Op packets by using the *ini* file parameter NoOpEnable. If No-Op packet transmission is activated, you can control the time interval in which No-Op packets are sent in the case of silence (i.e., no RTP or T.38 traffic). This is performed using the *ini* file parameter NoOpInterval. For a description of the RTP No-Op *ini* file parameters, refer to "Networking Parameters" on page 247.

- **RTP No-Op:** The RTP No-Op support complies with IETF's draft-wing-avt-rtp-noop-03.txt (titled 'A No-Op Payload Format for RTP'). This IETF document defines a No-Op payload format for RTP. The draft defines the RTP payload type as dynamic. You can control the payload type with which the No-Op packets are sent. This is performed using the RTPNoOpPayloadType *ini* parameter (refer to "Networking Parameters" on page 247). AudioCodes' default payload type is 120.
- **T.38 No-Op:** T.38 No-Op packets are sent only while a T.38 session is activated. Sent packets are a duplication of the previously sent frame (including duplication of the sequence number).

Note: Receipt of No-Op packets is always supported.

10.4 IP Multicasting

The device supports IP Multicasting level 1 according to RFC 2236 (i.e., IGMP version 2) for RTP channels. The device is capable of transmitting and receiving Multicast packets.

10.5 Robust Receipt of RTP Streams

This mechanism filters out unwanted RTP streams that are sent to the same port number on the device. These multiple RTP streams can result from traces of previous calls, call control errors, and deliberate attacks. When more than one RTP stream reaches the device on the same port number, the device accepts only one of the RTP streams and rejects the rest of the streams.

The RTP stream is selected according to the following: The first packet arriving on a newly opened channel sets the source IP address and UDP port from which further packets are received. Thus, the source IP address and UDP port identify the currently accepted stream. If a new packet arrives whose source IP address or UDP port are different to the currently accepted RTP stream, one of the following occurs:

- The device reverts to the new RTP stream when the new packet has a source IP address and UDP port that are the same as the remote IP address and UDP port that were stated during the opening of the channel.
- The packet is dropped when the new packet has any other source IP address and UDP port.

10.6 Multiple Routers Support

Multiple routers support is designed to assist the device when it operates in a multiple routers network. The device learns the network topology by responding to Internet Control Message Protocol (ICMP) redirections and caches them as routing rules (with expiration time).

When a set of routers operating within the same subnet serve as devices to that network and intercommunicate using a dynamic routing protocol, the routers can determine the shortest path to a certain destination and signal the remote host the existence of the better route. Using multiple router support, the device can utilize these router messages to change its next hop and establish the best path.

Note: Multiple Routers support is an integral feature that doesn't require configuration.

10.7 Simple Network Time Protocol Support

The Simple Network Time Protocol (SNTP) client functionality generates requests and reacts to the resulting responses using the NTP version 3 protocol definitions (according to RFC 1305). Through these requests and responses, the NTP client synchronizes the system time to a time source within the network, thereby eliminating any potential issues should the local system clock 'drift' during operation. By synchronizing time to a network time source, traffic handling, maintenance, and debugging become simplified for the network administrator.

The NTP client follows a simple process in managing system time: the NTP client requests an NTP update, receives an NTP response, and then updates the local system clock based on a configured NTP server within the network.

The client requests a time update from a specified NTP server at a specified update interval. In most situations, this update interval is every 24 hours based on when the system was restarted. The NTP server identity (as an IP address) and the update interval are user-defined (using the *ini* file parameters NTPServerIP and NTPUpdateInterval respectively), or an SNMP MIB object (refer to the *Product Reference Manual*).

When the client receives a response to its request from the identified NTP server, it must be interpreted based on time zone or location offset that the system is to a standard point of reference called the Universal Time Coordinate (UTC). The time offset that the NTP client uses is configurable using the *ini* file parameter NTPServerUTCOffset, or via an SNMP MIB object (refer to the *Product Reference Manual*).

If required, the clock update is performed by the client as the final step of the update process. The update is performed in such a way as to be transparent to the end users. For instance, the response of the server may indicate that the clock is running too fast on the client. The client slowly robs bits from the clock counter to update the clock to the correct time. If the clock is running too slow, then in an effort to catch the clock up, bits are added to the counter, causing the clock to update quicker and catch up to the correct time. The advantage of this method is that it does not introduce any disparity in the system time that is noticeable to an end user or that could corrupt call timeouts and timestamps.

10.8 IP QoS via Differentiated Services (DiffServ)

DiffServ is an architecture providing different types or levels of service for IP traffic. DiffServ (according to RFC 2474) offers the capability to prioritize certain traffic types depending on their priority, thereby, accomplishing a higher-level QoS at the expense of other traffic types. By prioritizing packets, DiffServ routers can minimize transmission delays for time-sensitive packets such as VoIP packets.

The device can be configured to set a different DiffServ value to IP packets according to their class-of-service: Network, Premium Media, Premium Control, Gold, and Bronze. The DiffServ parameters are described in "Networking Parameters" on page 247.

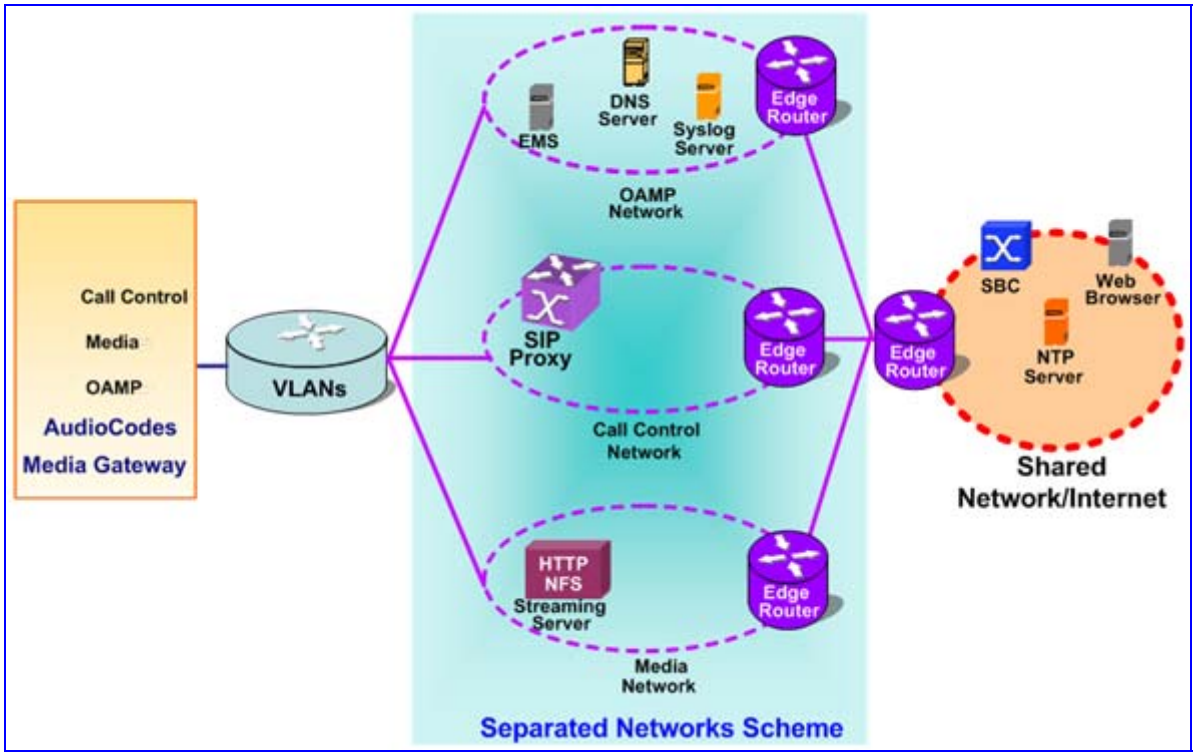
10.9 Network Configuration

The device allows you to configure up to 16 different IP addresses with associated VLANs, using the Multiple Interface table. In addition, complementing this table is the Routing table, which allows you to define routing rules for non-local hosts/subnets. This section describes the various network configuration options offered by the device.

10.9.1 Multiple Network Interfaces and VLANs

A need often arises to have logically separated network segments for various applications (for administrative and security reasons). This can be achieved by employing Layer-2 VLANs and Layer 3 subnets.

Figure 10-2: Multiple Network Interfaces



This figure above depicts a typical configuration featuring in which the device is configured with three network interfaces for:

- Operations, Administration, Maintenance, and Provisioning (OAMP) applications
- Call Control applications
- Media

It is connected to a VLAN-aware switch, which is used for directing traffic from (and to) the device to three separated Layer-3 broadcast domains according to VLAN tags (middle pane).

The Multiple Interfaces scheme allows the configuration of up to 16 different IP addresses, each associated with a unique VLAN ID. The configuration is performed using the Multiple Interface table, which is configurable using the *ini* file, Web, and SNMP interfaces.

10.9.1.1 Overview of Multiple Interface Table

The Multiple Interfaces scheme allows you to define up to 16 different IP addresses and VLANs in a table format, as shown below:

Table 10-1: Multiple Interface Table

Index Mode	Application	Interface	IP Address	Prefix Length	Default Gateway	VLAN ID	Interface Name
0	OAMP	IPv4	10.31.174.50	16	0.0.0.0	4	ManagementIF
1	Control	IPv4	10.32.174.50	16	0.0.0.0	5	ControlIF
2	Media	IPv4	10.33.174.50	16	10.33.0.1	6	Media1IF
3	Media	IPv4	10.34.174.50	16	0.0.0.0	7	Media2IF
4	Media	IPv4	10.35.174.50	16	0.0.0.0	8	Media3IF
5	Media	IPv4	10.36.174.50	16	0.0.0.0	9	Media4IF
6	Media	IPv4	10.37.174.50	16	0.0.0.0	10	Media5IF
7	Media	IPv4	10.38.174.50	16	0.0.0.0	11	Media6IF
8	Media	IPv4	10.39.174.50	16	0.0.0.0	12	Media7IF
9	Media	IPv4	10.40.174.50	16	0.0.0.0	13	Media8IF
10	Media & Control	IPv4	10.41.174.50	16	0.0.0.0	14	Media9IF
11	Media	IPv4	10.42.174.50	16	0.0.0.0	15	Media10IF
12	Media	IPv4	10.43.174.50	16	0.0.0.0	16	Media11IF
13	Media	IPv4	10.44.174.50	16	0.0.0.0	17	Media12IF
14	Media	IPv4	10.45.174.50	16	0.0.0.0	18	Media13IF
15	Media & Control	IPv4	10.46.174.50	16	0.0.0.0	19	Media14IF

Complementing the network configuration are some VLAN-related parameters, determining if VLANs are enabled and the 'Native' VLAN ID (refer to the sub-sections below) as well as VLAN priorities and DiffServ values for the supported Classes Of Service (refer to "Quality of Service Parameters" on page 614).

10.9.1.2 Columns of the Multiple Interface Table

Each row of the table defines a logical IP interface with its own IP address, subnet mask (represented by Prefix Length), VLAN ID (if VLANs are enabled), name, and application types that are allowed on this interface. One of the interfaces may have a 'default gateway' definition. Traffic destined to a subnet which does not meet any of the routing rules (either local or static routes) are forwarded to this gateway (as long this application type is allowed on this interface). Refer to "Gateway Column" on page 612 for more details.

10.9.1.2.1 Index Column

This column holds the index of each interface. Possible values are 0 to 15. Each interface index must be unique.

10.9.1.2.2 Application Types Column

This column defines the types of applications that are allowed on this interface:

- OAMP – Operations, Administration, Maintenance and Provisioning applications such as Web, Telnet, SSH, SNMP
- CONTROL – Call Control Protocols (i.e., SIP)
- MEDIA – RTP streams of Voice
- Various combinations of the above mentioned types

The following table shows the possible values of this column and their descriptions:

Table 10-2: Application Types

Value	Description
0	OAMP: only OAMP applications are allowed on this interface.
1	MEDIA: only Media (RTP) are allowed on this interface.
2	CONTROL: only Call Control applications are allowed on this interface.
3	OAMP & MEDIA: only OAMP and Media (RTP) applications are allowed on this interface.
4	OAMP & CONTROL: only OAMP and Call Control applications are allowed on this interface.
5	MEDIA & CONTROL: only Media (RTP) and Call Control applications are allowed on this interface.
6	OAMP, MEDIA & CONTROL: all of the application types are allowed on this interface.

For valid configuration guidelines, refer to “Multiple Interface Table Configuration Summary and Guidelines” on page 616 for more information.

10.9.1.2.3 Interface Mode Column

The Interface Mode column determines the method that this interface uses to acquire its IP address. For IPv4 Manual IP Address assignment, use "IPv4 Manual" (10).

10.9.1.2.4 IP Address and Prefix Length Columns

These columns allow the user to configure an IPv4 IP address and its related subnet mask.

The Prefix Length column holds the Classless Inter-Domain Routing (CIDR)-style representation of a dotted-decimal subnet notation. The CIDR-style representation uses a suffix indicating the number of bits which are set in the dotted-decimal format, in other words, 192.168.0.0/16 is synonymous with 192.168.0.0 and a subnet 255.255.0.0 (Refer to http://en.wikipedia.org/wiki/Classless_Inter-Domain_Routing for more information).

This CIDR notation lists the number of '1' bits in the subnet mask. So, a subnet mask of 255.0.0.0 (when broken down to its binary format) is represented by a prefix length of 8 (11111111 00000000 00000000 00000000), and a subnet mask of 255.255.255.252 is represented by a prefix length of 30 (11111111 11111111 11111111 11111100).

Each interface must have its own address space. Two interfaces may not share the same address space, or even part of it. The IP address should be configured as a dotted-decimal notation.

For IPv4 interfaces, the prefix length values range from 0 to 31.

OAMP Interface Address when Booting using BootP/DHCP

When booting using BootP/DHCP protocols, an IP address is obtained from the server. This address is used as the OAMP address for this session, overriding the address configured using the Multiple Interface table. The address specified for OAMP applications in the table becomes available when booting from flash again. This allows the device to operate with a temporary address for initial management and configuration while retaining the address to be used for deployment.

10.9.1.2.5 Gateway Column

This column defines a default gateway for the device. For this reason, only one default gateway may be configured. The default gateway's address must be on the same subnet as the interface address. In addition, the default gateway can only be configured on one of the interfaces running Media traffic.

A separate routing table allows configuring additional routing rules. Refer to “Routing Table” on page 618 for more details.

Note: The default gateway configured in the example below (200.200.85.1) is available for the applications allowed on that interface (Media & Control). Outgoing management traffic (originating on interface 0) is never directed to this default gateway.

Table 10-3: Configured Default Gateway Example

Index	Application Type	Interface Mode	IP Address	Prefix Length	Gateway	VLAN ID	Interface Name
0	OAMP	IPv4 Manual	192.168.85.14	16	0.0.0.0	100	Mgmt
1	Media & Control	IPv4 Manual	200.200.85.14	24	200.200.85.1	200	CntrlMedia

A separate routing table allows configuring routing rules. Configuring the following routing rule enables OAMP applications to access peers on subnet 17.17.0.0 through the gateway 192.168.0.1.

Table 10-4: Separate Routing Table Example

Destination	Prefix Length	Subnet Mask	Gateway	Interface	Metric
17.17.0.0	16	-	192.168.0.1	0	1

Refer to the Routing Table sub-section for more details.

10.9.1.2.6 VLAN ID Column

This column defines the VLAN ID for each interface. When using VLANs, this column must hold a unique value for each interface of the same address family.

10.9.1.2.7 Interface Name Column

This column allows the configuration of a short string (up to 16 characters) to name this interface. This name is displayed in management interfaces (Web, CLI, and SNMP) and is used in the Media Realm table. This column must have a unique value for each interface (no two interfaces can have the same name) and must not be left blank.

10.9.1.3 Other Related Parameters

The Multiple Interface table allows you to configure interfaces and their related parameters such as their VLAN ID or the interface name. This section lists additional parameters complementing this table functionality.

10.9.1.3.1 Booting using DHCP

The *DHCPEnable* parameter enables the device to boot while acquiring an IP address from a DHCP server. Note that when using this method, Multiple Interface table/VLANs and other advanced configuration options are disabled.

10.9.1.3.2 Enabling VLANs

The Multiple Interface table's column "VLAN ID" assigns a VLAN ID to each of the interfaces. Incoming traffic tagged with this VLAN ID are channeled to the related interface, and outgoing traffic from that interface are tagged with this VLAN ID. When VLANs are required, the parameter should be set to 1. The default value for this parameter is 0 (disabled).

10.9.1.3.3 'Native' VLAN ID

A 'Native' VLAN ID is the VLAN ID to which untagged incoming traffic are assigned. Outgoing packets sent to this VLAN are sent only with a priority tag (VLAN ID = 0). When the 'Native' VLAN ID is equal to one of the VLAN IDs configured in the Multiple Interface table (and VLANs are enabled), untagged incoming traffic are considered as an incoming traffic for that interface. Outgoing traffic sent from this interface are sent with the priority tag (tagged with VLAN ID = 0). When the 'Native' VLAN ID is different from any value in the "VLAN ID" column in the Multiple Interface table, untagged incoming traffic are discarded and all the outgoing traffic are fully tagged.

The 'Native' VLAN ID is configurable using the VlanNativeVlanId parameter (refer to the Setting up your System sub-section below). The default value of the 'Native' VLAN ID is 1.

Note: If *VlanNativeVlanId* is not configured (i.e., its default value of 1 occurs), but one of the interfaces has a VLAN ID configured to 1, this interface is still related to the 'Native' VLAN. If you do not wish to have a 'Native' VLAN ID, and want to use VLAN ID 1, ensure that the value of the *VlanNativeVlanId* parameter is different than any VLAN ID in the table.

10.9.1.3.4 Quality of Service Parameters

The device allows you to specify values for Layer-2 and Layer-3 priorities, by assigning values to the following service classes:

- Network Service class – network control traffic (ICMP, ARP)
- Premium Media service class – used for RTP Media traffic
- Premium Control Service class – used for Call Control traffic
- Gold Service class – used for streaming applications
- Bronze Service class – used for OAMP applications

The Layer-2 Quality of Service parameters enables setting the values for the 3 priority bits in the VLAN tag of frames related to a specific service class (according to the IEEE 802.1p standard). The Layer-3 Quality of Service (QoS) parameters enables setting the values of the DiffServ field in the IP Header of the frames related to a specific service class. The following QoS parameters can be set:

Table 10-5: Quality of Service Parameters

Parameter	Description
Layer-2 Class Of Service Parameter (VLAN Tag Priority Field)	
VlanNetworkServiceClassPriority	Sets the priority for the Network service class content
VLANPremiumServiceClassMediaPriority	Sets the priority for the Premium service class content (media traffic)
VLANPremiumServiceClassControlPriority	Sets the priority for the Premium service class content (control traffic)
VLANGoldServiceClassPriority	Sets the priority for the Gold service class content (streaming traffic)
VLANBronzeServiceClassPriority	Sets the priority for the Bronze service class content (OAMP traffic)
Layer-3 Class Of Service Parameters (TOS/DiffServ)	
NetworkServiceClassDiffServ	Sets the DiffServ for the Network service class content
PremiumServiceClassMediaDiffServ	Sets the DiffServ for the Premium service class content (media traffic)
PremiumServiceClassControlDiffServ	Sets the DiffServ for the Premium service class content (control traffic)
GoldServiceClassDiffServ	Sets the DiffServ for the Gold service class content (streaming traffic)
BronzeServiceClassDiffServ	Sets the DiffServ for the Bronze service class content (OAMP traffic)

The mapping of an application to its CoS and traffic type is shown in the table below:

Table 10-6: Traffic / Network Types and Priority

Application	Traffic / Network Types	Class-of-Service (Priority)
Debugging interface	Management	Bronze
Telnet	Management	Bronze
DHCP	Management	Network
Web server (HTTP)	Management	Bronze
SNMP GET/SET	Management	Bronze
Web server (HTTPS)	Management	Bronze
IPSec IKE	Determined by the service	Determined by the service
RTP traffic	Media	Premium media
RTCP traffic	Media	Premium media
T.38 traffic	Media	Premium media
SIP	Control	Premium control
SIP over TLS (SIPS)	Control	Premium control
Syslog	Management	Bronze
ICMP	Management	Determined by the initiator of the request
ARP listener	Determined by the initiator of the request	Network
SNMP Traps	Management	Bronze
DNS client	DNS (EnableDNSasOAM)	Network
NTP	NTP (EnableNTPasOAM)	Depends on traffic type: ▪ Control: Premium control ▪ Management: Bronze
NFS	NFSServers_VlanType in the NFSServers table	Gold

10.9.1.3.5 Applications with Assignable Application Type

Some applications can be associated with different application types in different setups. These application types are configurable. The applications listed below can be configured to one of two application types:

- DNS
- NTP

Table 10-7: Application Type Parameters

Parameter	Description
EnableDNSasOAM	This parameter applies to both Multiple IPs and VLAN mechanisms. Multiple IPs: Determines the network type for DNS services. VLAN: Determines the traffic type for DNS services. ▪ **[1]** = OAMP (default) ▪ **[0]** = Control. **Note:** For this parameter to take effect, a device reset is required.
EnableNTPasOAM	This parameter applies to both Multiple IPs and VLAN mechanisms. Multiple IPs: Determines the network type for NTP services. VLAN: Determines the traffic type for NTP services. ▪ **[1]** = OAMP (default) ▪ **[0]** = Control. **Note:** For this parameter to take effect, a device reset is required.

10.9.1.4 Multiple Interface Table Configuration Summary and Guidelines

Multiple Interface table configuration must adhere to the following rules:

- Up to 16 different interfaces may be defined.
- The indices used must be in the range between 0 to 15.
- Each interface must have its own subnet. Defining two interfaces with addresses in the same subnet (i.e. two interfaces with 192.168.0.1/16 and 192.168.100.1/16) is illegal.
- Subnets in different interfaces must not be overlapping in any way (i.e. defining two interfaces with 10.0.0.1/8 and 10.50.10.1/24 is invalid). Each interface must have its own address space.
- The Prefix Length replaces the dotted decimal Subnet Mask presentation. This column must have a value of 0-31 for IPv4 interfaces.
- Only one IPv4 interface with OAMP "Application Types" **must** be configured. At least one IPv4 interface with CONTROL "Application Types" **must** be configured. At least one IPv4 interface with MEDIA "Application Types" **must** be configured. These application types **may** be mixed (i.e. OAMP and CONTROL). Here are some examples for interface configuration:
  - One IPv4 interface with "Application Types" OAMP, MEDIA & CONTROL (without VLANs).
  - One IPv4 interface with "Application Types" OAMP, MEDIA & CONTROL.
  - One IPv4 interface with "Application Types" OAMP, one other or more IPv4 interfaces with "Application Types" CONTROL, and one or more IPv4 interfaces with "Application Types" MEDIA (with VLANs).
  - One IPv4 interface with "Application Types" OAMP & MEDIA, one other or more IPv4 interfaces with "Application Types" MEDIA & CONTROL.
  - Other configurations are also possible while keeping to the above-mentioned rule.
- Only one interface may have a Gateway definition for each address family (IPv4). This Gateway address must be in the same subnet as this interface; other routing rules may be specified in the Routing Table. Refer to “Routing Table” on page 618 for more details.

- Apart from the interface having the default gateway defined, the Gateway column for all other interfaces must be set to "0.0.0.0" for IPv4.
- The Interface Name column may have up to 16 characters. This column allows the user to name each interface with an easier name to associate the interface with. This column must have a unique value to each interface and must not be left blank.
- For IPv4 interfaces, the "Interface Mode" column must be set to "IPv4 Manual" (numeric value 10).
- When defining more than one interface of the same address family, VLANs must be enabled (the VlanMode should be set to 1).
- VLANs become available only when booting the device from flash. When booting using BootP/DHCP protocols, VLANs are disabled to allow easier maintenance access. In this scenario, multiple network interface capabilities are not available.
- The 'Native' VLAN ID may be defined using the 'VlanNativeVlanId' parameter. This relates untagged incoming traffic as if reached with a specified VLAN ID. Outgoing traffic from the interface which VLAN ID equals to the 'Native' VLAN ID are tagged with VLAN ID 0 (priority tag).
- Quality of Service parameters specify the priority field for the VLAN tag (IEEE 802.1p) and the DiffServ field for the IP headers. These specifications relate to service classes.
- When booting using BootP/DHCP protocols, the address received from the BootP/DHCP server acts as a temporary OAMP address, regardless of the address specified in the Multiple Interface table. This configured address becomes available when booting from flash.
- Network Configuration changes are offline. The new configuration should be saved and becomes available at the next startup.

Upon system start up, the Multiple Interface table is parsed and passes comprehensive validation tests. If any errors occur during this validation phase, the device sends an error message to the Syslog server and falls back to a "safe mode", using a single interface and no VLANs. Please be sure to follow the Syslog messages that the device sends in system startup to see if any errors occurred.

Note: When configuring the device using the Web interface, it is possible to perform a quick validation of the configured Multiple Interface table and VLAN definitions, by clicking the **Done** button in the Multiple Interface Table Web page. It is highly recommended to perform this when configuring Multiple Interfaces and VLANs, using the Web Interface to ensure the configuration is complete and valid.

10.9.1.5 Troubleshooting the Multiple Interface Table

If any of the Multiple Interface table guidelines are violated, the device falls back to a "safe mode" configuration, consisting of a single IPv4 interface and no VLANs. For more information on validation failures, consult the Syslog messages.

Validation failures may be caused by one of the following:

- One of the Application Types (OAMP, CONTROL, MEDIA) is missing in the IPv4 interfaces.
- There are too many interfaces with "Application Types" of OAMP. Only one interface defined but the "Application Types" column is not set to "O+M+C" (numeric value 6).

- An IPv4 interface was defined with "Interface Type" different than "IPv4 Manual" (10).
- Gateway column is filled in more than one row of the same address family.
- Gateway is defined in an interface not having MEDIA as one of its "Application Types".
- Two interfaces have the exact VLAN ID value, while VLANs are enabled.
- Two interfaces have the same name.
- Two interfaces share the same address space or subnet.

Apart from these validation errors, connectivity problems may be caused by one of the following:

- Trying to access the device with VLAN tags while booting from BootP/DHCP.
- Trying to access the device with untagged traffic when VLANs are on and Native VLAN is not configured properly.
- Routing Table is not configured properly.

10.9.2 Routing Table

The routing table allows you to configure routing rules. You may define up to 25 different routing rules, using the *ini* file, Web interface, and SNMP.

10.9.2.1 Routing Table Overview

The Routing Table consists of the following:

Table 10-8: Routing Table Layout

Destination	Prefix Length	Subnet Mask	Gateway	Interface	Metric
201.201.0.0	16	255.255.0.0	192.168.0.1	0	1
202.202.0.0	16	255.255.0.0	192.168.0.2	0	1
203.203.0.0	16	255.255.0.0	192.168.0.3	0	1
225.225.0.0	16	255.255.0.0	192.168.0.25	0	1

10.9.2.2 Routing Table Columns

Each row of the Routing table defines a routing rule. Traffic destined to the subnet specified in the routing rule is re-directed to a specified gateway, reachable through a specified interface.

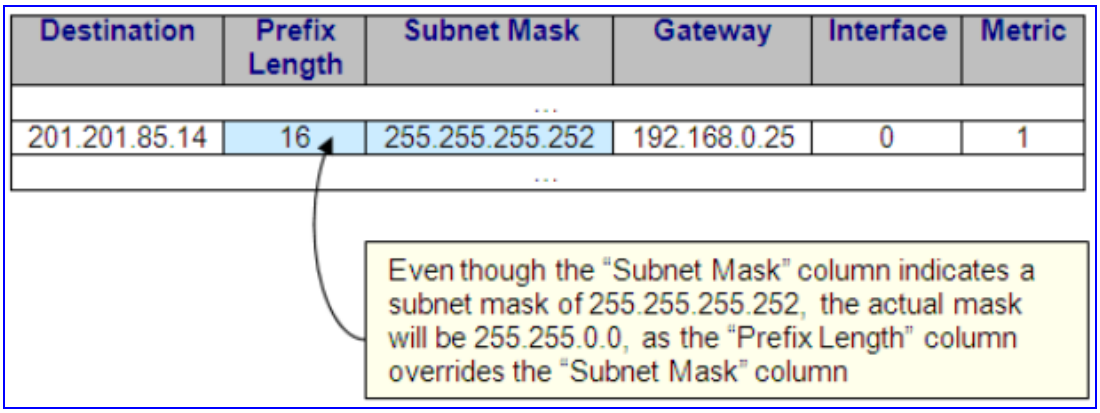
10.9.2.2.1 Destination Column

This column defines the destination of the route rule. The destination can be a single host or a whole subnet, depending on the Prefix Length/Subnet Mask specified for this routing rule.

10.9.2.2.2 Prefix Length and Subnet Mask Columns

These two columns offer two notations for the mask. You can either enable the Subnet Mask in dotted-decimal notation, or the CIDR-style representation. Please note that only one of these is needed. If both are specified, the "Prefix Length" column overrides the "Subnet Mask" column.

Figure 10-3: Prefix Length and Subnet Masks Columns



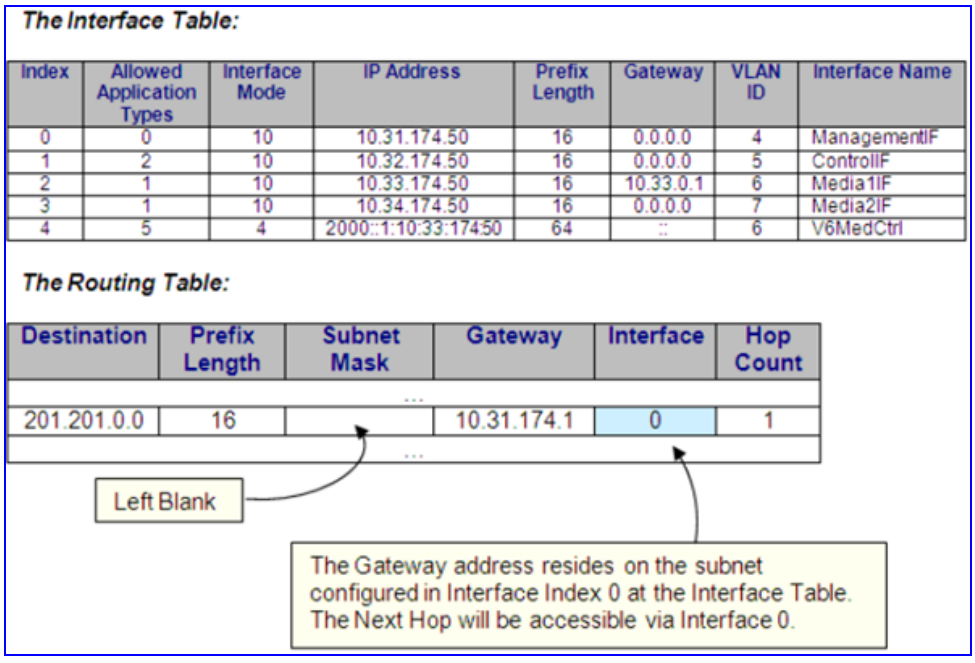
10.9.2.2.3 Gateway Column

The Gateway column defines the IP Address of the next hop used for traffic, destined to the subnet, as specified by the destination/mask columns. This gateway address must be on one of the subnets on which the address is configured in the Multiple Interface table.

10.9.2.2.4 Interface Column

This column defines the interface index (in the Multiple Interface table) from which the gateway address is reached.

Figure 10-4: Interface Column


The Interface Table:
Index
Allowed Application Types
Interface Mode
IP Address
Prefix Length
Gateway
VLAN ID
Interface Name
0 0 10 10.31.174.50 16 0.0.0.0 4 ManagementIF
1 2 10 10.32.174.50 16 0.0.0.0 5 ControlIF
2 1 10 10.33.174.50 16 10.33.0.1 6 Media1IF
3 1 10 10.34.174.50 16 0.0.0.0 7 Media2IF
4 5 4 2000::1:10:33:17450 64 :: 6 V6MedCtrl
The Routing Table:
Destination
Prefix Length
Subnet Mask
Gateway
Interface
Hop Count
201.201.0.0 16 10.31.174.1 0 1
Left Blank
The Gateway address resides on the subnet configured in Interface Index 0 at the Interface Table. The Next Hop will be accessible via Interface 0.

10.9.2.2.5 Metric Column

The Metric column must be set to 1 for each routing rule.

10.9.2.3 Routing Table Configuration Summary and Guidelines

The Routing table configurations must adhere to the following rules:

- Up to 25 different routing rules may be defined.
- The user may choose whether to specify "Prefix Length" or "Subnet Mask". There is no need to specify both.
- If both "Prefix Length" and "Subnet Mask" are defined, the "Prefix Length" overrides the "Subnet Mask".
- The "Gateway" IP Address must be available on one of the local subnets.
- The "Interface" column must be set to the Interface that the "Gateway" is configured on.
- The "Metric" column must be set to 1.
- The Routing Table configuration, unlike the Multiple Interface table configuration is online. Therefore, the changes made to the routing rules are applied immediately.

10.9.2.4 Troubleshooting the Routing Table

When adding or modifying any of the routing rules, the added or modified rule passes a validation test. If errors are found, the route is rejected and is not added to the Routing table. Failed routing validations may result in limited connectivity (or no connectivity) to the destinations specified in the incorrect routing rule. For any error found in the Routing table or failure to configure a routing rule, the device sends a notification message to the Syslog server reporting the problem.

Common routing rule configuration errors may include the following:

- The IP address specified in the "Gateway" column is unreachable from the interface specified in the "Interface" column.
- The same destination is defined in two different routing rules.
- "Subnet Mask" and "Prefix Length" columns are both entered with inconsistent values, and the "Prefix Length" overrides the "Subnet Mask" column.
- More than 25 routing rules were specified.

Note: If a routing rule is required to access OAMP applications (for remote management, for instance) and this route is not configured correctly, the route is not added and the device is not accessible remotely. To restore connectivity, the device must be accessed locally from the OAMP subnet and the required routes be configured.

10.9.3 Setting up the Device

10.9.3.1 Using the Web Interface

The Web interface is a convenient user interface for configuring the device's network configuration.

10.9.3.2 Using the *ini* File

When configuring the network configuration using the *ini* File, use a textual presentation of the Interface and Routing Tables, as well as some other parameters.

The following shows an example of a full network configuration, consisting of **all** the parameters described in this section.

```
; VLAN related parameters:
VlanMode = 0
VlanNativeVlanId = 1
; Routing Table Configuration:
RoutingTableDestinationsColumn = 201.201.0.0, 202.202.0.0
RoutingTableDestinationPrefixLensColumn = 16, 16
RoutingTableGatewaysColumn = 192.168.0.2, 192.168.0.3
RoutingTableInterfacesColumn = 0, 0
RoutingTableHopsCountColumn = 1, 1
; Class Of Service parameters:
VlanNetworkServiceClassPriority = 7
VlanPremiumServiceClassMediaPriority = 6
VlanPremiumServiceClassControlPriority = 6
VlanGoldServiceClassPriority = 4
VlanBronzeServiceClassPriority = 2
NetworkServiceClassDiffServ = 48
PremiumServiceClassMediaDiffServ = 46
PremiumServiceClassControlDiffServ = 40
GoldServiceClassDiffServ = 26
BronzeServiceClassDiffServ = 10
; Application Type for applications:
EnableDNSasOAM = 1
EnableNTPasOAM = 1
; Multiple Interface Table Configuration:
[InterfaceTable]
FORMAT InterfaceTable Index = InterfaceTable ApplicationTypes,
InterfaceTable InterfaceMode, InterfaceTable IPAddress,
InterfaceTable_PrefixLength, InterfaceTable_Gateway,
InterfaceTable VlanID, InterfaceTable InterfaceName;
InterfaceTable 0  = 6, 10, 192.168.85.14,  16, 192.168.0.1, 1, myAll;
```

This *ini* file shows the following:

- A Multiple Interface table with a single interface (192.168.85.14/16, OAMP, Media and Control applications are allowed) and a default gateway (192.168.0.1).
- A Routing table is configured with two routing rules, directing all traffic for subnet 201.201.0.0/16 to 192.168.0.2, and all traffic for subnet 202.202.0.0/16 to 192.168.0.3.
- VLANs are disabled, 'Native' VLAN ID is set to 1.
- Values for the Class Of Service parameters are assigned.

- The DNS application is configured to act as an OAMP application and the NTP application is configured to act as an OAMP application.

Notes:

- Lines that begin with a semicolon are considered a remark and are ignored.
- The Multiple Interface table configuration using the *ini* file must have the prefix and suffix to allow AudioCodes INI File parser to correctly recognize the Multiple Interface Table.

The following sections show some examples of selected network configurations, and their matching *ini* file configuration.

Example 1: Single Interface Configuration - Multiple Interface table with a single interface for OAMP, Media and Control applications:

Table 10-9: Multiple Interface Table - Example1

Index	Application	Interface	IP Address	Prefix Length	Default Gateway	VLAN ID	Interface Name
0	OAMP, Media & Control	IPv4	192.168.85.14	16	192.168.0.1	1	myInterface

VLANS are not required and the 'Native' VLAN ID is irrelevant. Class of Service parameters may have default values. The required routing table features two routes:

Table 10-10: Routing Table - Example 1

Destination	Prefix Length	Subnet Mask	Gateway	Interface	Metric
201.201.0.0	16		192.168.0.2	0	1
202.202.0.0	16		192.168.0.3	0	1

The DNS/NTP applications may have their default application types. This example's matching *ini* file is shown above. However, since many parameter values equal their default values, they can be omitted. The *ini* file can be also written as follows:

```
; Interface Table Configuration:
[InterfaceTable]
FORMAT InterfaceTable Index = InterfaceTable ApplicationTypes,
InterfaceTable InterfaceMode, InterfaceTable IPAddress,
InterfaceTable PrefixLength, InterfaceTable Gateway,
InterfaceTable VlanID, InterfaceTable InterfaceName;
InterfaceTable 0  = 6, 10, 192.168.85.14,  16, 192.168.0.1, 1, myAll;
[\InterfaceTable]
; Routing Table Configuration:
RoutingTableDestinationsColumn = 201.201.0.0, 202.202.0.0
RoutingTableDestinationPrefixLensColumn = 16, 16
RoutingTableGatewaysColumn = 192.168.0.2, 192.168.0.3
RoutingTableInterfacesColumn = 0, 0
RoutingTableHopsCountColumn = 1, 1
```

Example 2: Three Interfaces, one for each application exclusively - the Multiple Interface table is configured with three interfaces, one exclusively for each application type: one interface for OAMP applications, one for Call Control applications, and one for RTP Media applications:

Table 10-11: Multiple Interface Table - Example 2

Index	Application	Interface	IP Address	Prefix Length	Default Gateway	VLAN ID	Interface Name
0	OAMP	IPv4	192.168.85.14	16	0.0.0.0	1	ManagementIF
1	Control	IPv4	200.200.85.14	24	0.0.0.0	200	myControlIF
2	Media	IPv4	211.211.85.14	24	211.211.85.1	211	myMediaIF

VLANs are required. The 'Native' VLAN ID is the same VLAN ID as the Management interface (Index 0). One routing rule is required to allow remote management from a host in 176.85.49.0 / 24:

Table 10-12: Routing Table - Example 2

Destination	Prefix Length	Subnet Mask	Gateway	Interface	Metric
176.85.49.0	24		192.168.0.1	0	1

All other parameters are set to their respective default values. The *ini* file matching this configuration can be written as follows:

```
; Interface Table Configuration:
[InterfaceTable]
FORMAT InterfaceTable Index = InterfaceTable ApplicationTypes,
InterfaceTable InterfaceMode, InterfaceTable IPAddress,
InterfaceTable_PrefixLength, InterfaceTable_Gateway,
InterfaceTable VlanID, InterfaceTable InterfaceName;
InterfaceTable 0  = 0, 10, 192.168.85.14,  16, 0.0.0.0, 1, ManagementIF;
InterfaceTable 1  = 2, 10, 200.200.85.14,  24, 0.0.0.0, 200, myControlIF;
InterfaceTable 2  = 1, 10, 211.211.85.14,  24, 211.211.85.1, 211,
myMediaIF;
[\InterfaceTable]
; VLAN related parameters:
VlanMode = 1
VlanNativeVlanId = 1
; Routing Table Configuration:
RoutingTableDestinationsColumn = 176.85.49.0
RoutingTableDestinationPrefixLensColumn = 24
RoutingTableGatewaysColumn = 192.168.0.1
RoutingTableInterfacesColumn = 0
RoutingTableHopsCountColumn = 1
```

Example 3 - One interface exclusively for management (OAMP applications) and two others for Call Control and RTP (CONTROL and MEDIA applications):

The Multiple Interface table is configured with four interfaces. One is exclusively for Management and the two are for Call Control and RTP Media applications. Two of them are IPv4 interfaces:

Table 10-13: Multiple Interface Table - Example 3

Index	Application	Interface	IP Address	Prefix Length	Default Gateway	VLAN ID	Interface Name
0	OAMP	IPv4	192.168.85.14	16	0.0.0.0	1	Mgmt
1	Media & Control	IPv4	200.200.85.14	24	200.200.85.1	201	CntrlMedia1
2	Media & Control	IPv4	200.200.86.14	24	0.0.0.0	202	CntrlMedia2

VLANs are required. The 'Native' VLAN ID is the same VLAN ID as the AudioCodes Management interface (index 0). One routing rule is required to allow remote management from a host in 176.85.49.0/24.

Table 10-14: Routing Table - Example 3

Destination	Prefix Length	Subnet Mask	Gateway	Interface	Metric
176.85.49.0	24		192.168.0.1	0	1

All other parameters are set to their respective default values. The *ini* file matching this configuration can be written as follows:

```
; Interface Table Configuration:
[InterfaceTable]
FORMAT InterfaceTable Index = InterfaceTable ApplicationTypes,
InterfaceTable InterfaceMode, InterfaceTable IPAddress,
InterfaceTable_PrefixLength, InterfaceTable_Gateway,
InterfaceTable VlanID, InterfaceTable InterfaceName;
InterfaceTable 0  = 0, 10, 192.168.85.14,  16, 0.0.0.0, 1, Mgmt;
InterfaceTable 1  = 5, 10, 200.200.85.14,  24, 200.200.85.1, 201,
CntrlMedia1;
InterfaceTable 2  = 5, 10, 200.200.86.14,  24, 0.0.0.0, 202, CntrlMedia2;

[\InterfaceTable]
; VLAN related parameters:
VlanMode = 1
VlanNativeVlanId = 1
; Routing Table Configuration:
RoutingTableDestinationsColumn = 176.85.49.0
RoutingTableDestinationPrefixLensColumn = 24
RoutingTableGatewaysColumn = 192.168.0.1
RoutingTableInterfacesColumn = 0
RoutingTableHopsCountColumn = 1
```

11 Advanced PSTN Configuration

This section discusses advanced PSTN configurations.

11.1 Clock Settings

In a traditional TDM service network such as PSTN, both ends of the TDM connection must be synchronized. If synchronization is not achieved, voice frames are either dropped (to prevent a buffer overflow condition) or inserted (to prevent an underflow condition). In both cases, connection quality and reliability is affected.

The device's clock settings can be configured to one of the following:

- **Network:** recovers a clock from one of its physical PSTN E1/T1 trunk interfaces as a clock source for transmit on all interfaces
- **Internal/BITS:** uses the internal oscillator for its transmit

Note: The device terminates the E1/T1 down to 64-kbps timeslots and hence, can't tolerate different clock sources for its physical interfaces. Different clock sources can lead to performance errors and voice disturbances.

➢ **To use the device's internal clock source,:**

1. TDMBusClockSource = 1
2. ClockMaster = 1 (for all trunks)

➢ **To use the recovered clock option:**

1. TDMBusClockSource = 4
2. ClockMaster_x = 0 (for all 'slave' trunks connected to PBX#1)
3. ClockMaster_x = 1 (for all 'master' trunks connected to PBX#2)

The above assumes that the device recovers its internal clock from one of the 'slave' trunks connected to PBX#1 and provides clock to PBX#2 on its 'master' trunks. In addition, it's necessary to define from which of the 'slave' trunks the device recovers its clock:

- TDMBusPSTNAutoClockEnable = 1 (device automatically selects one of the connected 'slave' trunks)
  - Or -
- TDMBusLocalReference = # (trunk number, where 0 is the first trunk - and the default)

Note: When the device is used in a 'non-span' configuration, the internal device clock must be used (as explained above).

11.2 Release Reason Mapping

This section describes the available mapping mechanisms of SIP responses to Q.850 Release Causes and vice versa. The existing mapping of ISDN Release Causes to SIP Responses is described in "Fixed Mapping of ISDN Release Reason to SIP Response" on page 626 and "Fixed Mapping of SIP Response to ISDN Release Reason" on page 628. To override this hard-coded mapping and flexibly map SIP responses to ISDN Release Causes, use the *ini* file (CauseMapISDN2SIP and CauseMapSIP2ISDN, as described in "ISDN and CAS Interworking Parameters" on page 392) or the Web interface (refer to "Configuring Release Cause Mapping" on page 169).

It is also possible to map the less commonly used SIP responses to a single default ISDN Release Cause. Use the parameter DefaultCauseMapISDN2IP (described in "ISDN and CAS Interworking Parameters" on page 392) to define a default ISDN Cause that is always used except when the following Release Causes are received: Normal Call Clearing (16), User Busy (17), No User Responding (18) or No Answer from User (19). This mechanism is only available for Tel-to-IP calls.

11.2.1 Reason Header

The device supports the Reason header according to RFC 3326. The Reason header conveys information describing the disconnection cause of a call:

- **Sending Reason header:** If a call is disconnected from the Tel side (ISDN), the Reason header is set to the received Q.850 cause in the appropriate message (BYE/CANCEL/final failure response) and sent to the SIP side. If the call is disconnected because of a SIP reason, the Reason header is set to the appropriate SIP response.
- **Receiving Reason header:** If a call is disconnected from the IP side and the SIP message includes the Reason header, it is sent to the Tel side according to the following logic:
  - If the Reason header includes a Q.850 cause, it is sent as is.
  - If the Reason header includes a SIP response:
    - If the message is a final response, the response status code is translated to Q.850 format and passed to ISDN.
    - If the message isn't a final response, it is translated to a Q.850 cause.
  - When the Reason header is received twice (i.e., SIP Reason and Q.850), the Q.850 takes precedence over the SIP reason and is sent to the Tel side.

11.2.2 Fixed Mapping of ISDN Release Reason to SIP Response

The following table describes the mapping of ISDN release reason to SIP response.

Table 11-1: Mapping of ISDN Release Reason to SIP Response

ISDN Release Reason	Description	SIP Response	Description
1	Unallocated number	404	Not found
2	No route to network	404	Not found
3	No route to destination	404	Not found

ISDN Release Reason	Description	SIP Response	Description
6	Channel unacceptable	406*	Not acceptable
7	Call awarded and being delivered in an established channel	500	Server internal error
16	Normal call clearing	-*	BYE
17	User busy	486	Busy here
18	No user responding	408	Request timeout
19	No answer from the user	480	Temporarily unavailable
21	Call rejected	403	Forbidden
22	Number changed w/o diagnostic	410	Gone
26	Non-selected user clearing	404	Not found
27	Destination out of order	502	Bad gateway
28	Address incomplete	484	Address incomplete
29	Facility rejected	501	Not implemented
30	Response to status enquiry	501*	Not implemented
31	Normal unspecified	480	Temporarily unavailable
34	No circuit available	503	Service unavailable
38	Network out of order	503	Service unavailable
41	Temporary failure	503	Service unavailable
42	Switching equipment congestion	503	Service unavailable
43	Access information discarded	502*	Bad gateway
44	Requested channel not available	503*	Service unavailable
47	Resource unavailable	503	Service unavailable
49	QoS unavailable	503*	Service unavailable
50	Facility not subscribed	503*	Service unavailable
55	Incoming calls barred within CUG	403	Forbidden
57	Bearer capability not authorized	403	Forbidden
58	Bearer capability not presently available	503	Service unavailable
63	Service/option not available	503*	Service unavailable
65	Bearer capability not implemented	501	Not implemented
66	Channel type not implemented	480*	Temporarily unavailable
69	Requested facility not implemented	503*	Service unavailable
70	Only restricted digital information bearer capability is available	503*	Service unavailable
79	Service or option not implemented	501	Not implemented

ISDN Release Reason	Description	SIP Response	Description
81	Invalid call reference value	502*	Bad gateway
82	Identified channel does not exist	502*	Bad gateway
83	Suspended call exists, but this call identity does not	503*	Service unavailable
84	Call identity in use	503*	Service unavailable
85	No call suspended	503*	Service unavailable
86	Call having the requested call identity has been cleared	408*	Request timeout
87	User not member of CUG	503	Service unavailable
88	Incompatible destination	503	Service unavailable
91	Invalid transit network selection	502*	Bad gateway
95	Invalid message	503	Service unavailable
96	Mandatory information element is missing	409*	Conflict
97	Message type non-existent or not implemented	480*	Temporarily not available
98	Message not compatible with call state or message type non-existent or not implemented	409*	Conflict
99	Information element non-existent or not implemented	480*	Not found
100	Invalid information elements contents	501*	Not implemented
101	Message not compatible with call state	503*	Service unavailable
102	Recovery of timer expiry	408	Request timeout
111	Protocol error	500	Server internal error
127	Interworking unspecified	500	Server internal error

***** Messages and responses were created because the 'ISUP to SIP Mapping' draft doesn't specify their cause code mapping.

11.2.3 Fixed Mapping of SIP Response to ISDN Release Reason

The following table describes the mapping of SIP response to ISDN release reason.

Table 11-2: Mapping of SIP Response to ISDN Release Reason

SIP Response	Description	ISDN Release Reason	Description
400*	Bad request	31	Normal, unspecified
401	Unauthorized	21	Call rejected

SIP Response	Description	ISDN Release Reason	Description
402	Payment required	21	Call rejected
403	Forbidden	21	Call rejected
404	Not found	1	Unallocated number
405	Method not allowed	63	Service/option unavailable
406	Not acceptable	79	Service/option not implemented
407	Proxy authentication required	21	Call rejected
408	Request timeout	102	Recovery on timer expiry
409	Conflict	41	Temporary failure
410	Gone	22	Number changed w/o diagnostic
411	Length required	127	Interworking
413	Request entity too long	127	Interworking
414	Request URI too long	127	Interworking
415	Unsupported media type	79	Service/option not implemented
420	Bad extension	127	Interworking
480	Temporarily unavailable	18	No user responding
481*	Call leg/transaction doesn't exist	127	Interworking
482*	Loop detected	127	Interworking
483	Too many hops	127	Interworking
484	Address incomplete	28	Invalid number format
485	Ambiguous	1	Unallocated number
486	Busy here	17	User busy
488	Not acceptable here	31	Normal, unspecified
500	Server internal error	41	Temporary failure
501	Not implemented	38	Network out of order
502	Bad gateway	38	Network out of order
503	Service unavailable	41	Temporary failure
504	Server timeout	102	Recovery on timer expiry
505*	Version not supported	127	Interworking
600	Busy everywhere	17	User busy
603	Decline	21	Call rejected
604	Does not exist anywhere	1	Unallocated number
606*	Not acceptable	38	Network out of order

***** Messages and responses were created because the 'ISUP to SIP Mapping' draft does not specify their cause code mapping.

11.3 ISDN Overlap Dialing

Overlap dialing is a dialing scheme used by several ISDN variants to send and/or receive called number digits one after the other (or several at a time). This is in contrast to en-bloc dialing in which a complete number is sent.

The device now supports the interworking of ISDN overlap dialing to SIP, based on RFC 3578.

- **Interworking ISDN overlap dialing to SIP (Tel to IP):** The device sends collected digits each time they are received (initially from ISDN Setup and then from subsequent Info Q.931 messages) to the IP, using subsequent SIP INVITE messages. You can also define the minimum number of overlap digits to collect before sending the first SIP message (INVITE) for routing the call, using the new parameter MinOverlapDigitsForRouting.

  The device stops collecting digits when:

  - Receives Sending Complete IE in the ISDN Setup or Info messages to signal that no more digits are going to be sent.
  - The inter-digit timeout (configured by the parameter TimeBetweenDigits) expires.
  - The maximum allowed number of digits (configured by the parameter MaxDigits) is reached.
  - A match is found with the defined digit map (configured by the parameter DigitMapping).

  The device can also mute in-band DTMF detection until the device receives the full destination number from the ISDN. This is configured using the MuteDTMFInOverlap parameter. With ISDN overlap dialing, DTMF digits can be sent in-band in the voice stream or out-of-band using Q.931 Info messages. If Q.931 Info messages are used, the DTMF in-band detector must be disabled. Note that when at least one digit is received from the ISDN (Setup message), the device stops playing a dial tone.

- **Interworking SIP to ISDN overlap dialing (IP to Tel):** For each received INVITE pertaining to the same dialog session, the device sends an ISDN Setup (and subsequent ISDN Info Q.931 messages) with the collected digits to the Tel side. For all subsequent INVITEs received, the device sends a SIP 484 "Address Incomplete" response to the IP in order to maintain the current dialog session and to receive additional digits from subsequent INVITEs.

The device can optionally support ISDN overlap dialing for incoming ISDN calls for the entire device or per E1/T1 span, using the ISDNRxOverlap parameter.

To play a Dial tone to the ISDN user side when an empty called number is received, set the parameter ISDNINCallsBehavior to 65536 (bit #16). This results in the Progress Indicator being included in the SetupAck ISDN message.

Relevant parameters (described in "PSTN Parameters" on page 375):

- ISDNRxOverlap
- ISDNTxOverlap
- TimeBetweenDigits
- MaxDigits
- ISDNInCallsBehavior

- DigitMapping
- MinOverlapDigitsForRouting

For configuring ISDN overlap dialing using the Web interface, refer to "Configuring the Trunk Settings" on page 80.

11.4 ISDN Non-Facility Associated Signaling (NFAS)

In regular T1 ISDN trunks, a single 64 kbps channel carries signaling for the other 23 B-channels of that particular T1 trunk. This channel is called the D-channel and usually resides on timeslot # 24. The ISDN Non-Facility Associated Signaling (NFAS) feature enables the use of a single D-channel to control multiple PRI interfaces.

With NFAS it is possible to define a group of T1 trunks, called an NFAS group, in which a single D-channel carries ISDN signaling messages for the entire group. The NFAS group's B-channels are used to carry traffic such as voice or data. The NFAS mechanism also enables definition of a backup D-channel on a different T1 trunk, to be used if the primary D-channel fails.

The NFAS group can comprise up to 10 T1 trunks. Each T1 trunk is called an 'NFAS member'. The T1 trunk whose D-channel is used for signaling is called the 'Primary NFAS Trunk'. The T1 trunk whose D-channel is used for backup signaling is called the 'Backup NFAS Trunk'. The primary and backup trunks each carry 23 B-channels while all other NFAS trunks each carry 24 B-channels.

The device supports up to 9 NFAS groups. Each group must contain different T1 trunks.

The NFAS group is identified by an NFAS GroupID number (possible values are 1 to 9). To assign a number of T1 trunks to the same NFAS group, use the *ini* file parameter NFASGroupNumber_x = groupID (where *x* is the physical trunk ID (0 to the maximum number of trunks) or the Web interface (refer to "Configuring the Trunk Settings" on page 80).

The parameter 'DchConfig_x = Trunk_type' defines the type of NFAS trunk. Trunk_type is set to 0 for the primary trunk, to 1 for the backup trunk, and to 2 for an ordinary NFAS trunk. 'x' depicts the physical trunk ID (0 to the maximum number of trunks). You can also use the Web interface (refer to "Configuring the Trunk Settings" on page 80).

For example, to assign the first four T1 trunks to NFAS group #1, in which trunk #0 is the primary trunk and trunk #1 is the backup trunk, use the following configuration:

```
NFASGroupNumber 0 = 1
NFASGroupNumber 1 = 1
NFASGroupNumber 2 = 1
NFASGroupNumber_3 = 1
DchConfig 0 = 0              ;Primary T1 trunk
DchConfig 1 = 1              ;Backup T1 trunk
DchConfig 2 = 2              ;24 B-channel NFAS trunk
DchConfig_3 = 2              ;24 B-channel NFAS trunk
```

The NFAS parameters are described in "PSTN Parameters" on page 375.

11.4.1 NFAS Interface ID

Several ISDN switches require an additional configuration parameter per T1 trunk that is called 'Interface Identifier'. In NFAS T1 trunks, the Interface Identifier is sent explicitly in Q.931 Setup / Channel Identification IE for all NFAS trunks, except for the B-channels of the Primary trunk (refer to note below).

The Interface ID can be defined per member (T1 trunk) of the NFAS group, and must be coordinated with the configuration of the Switch. The default value of the Interface ID is identical to the number of the physical T1 trunk (0 for the first trunk, 1 for the second T1 trunk, and so on, up to the maximum number of trunks).

To define an explicit Interface ID for a T1 trunk (that is different from the default), use the following parameters:

- ISDNIBehavior_x = 512 (x = 0 to the maximum number of trunks identifying the device's physical trunk)
- ISDNNFASInterfaceID_x = ID (x = 0 to 255)

Notes:

- Usually the Interface Identifier is included in the Q.931 Setup/Channel Identification IE only on T1 trunks that doesn't contain the D-channel. Calls initiated on B-channels of the Primary T1 trunk, by default, don't contain the Interface Identifier. Setting the parameter ISDNIBehavior_x to 2048' forces the inclusion of the Channel Identifier parameter also for the Primary trunk.
- The parameter ISDNNFASInterfaceID_x = ID can define the 'Interface ID' for any Primary T1 trunk, even if the T1 trunk is not a part of an NFAS group. However, to include the Interface Identifier in Q.931 Setup/Channel Identification IE configure ISDNIBehavior_x = 2048 in the *ini* file.

11.4.2 Working with DMS-100 Switches

The DMS-100 switch requires the following NFAS Interface ID definitions:

- InterfaceID #0 for the Primary trunk
- InterfaceID #1 for the Backup trunk
- InterfaceID #2 for a 24 B-channel T1 trunk
- InterfaceID #3 for a 24 B-channel T1 trunk, and so on for subsequent T1 trunks

For example, if four T1 trunks on a device are configured as a single NFAS group with Primary and Backup T1 trunks that is used with a DMS-100 switch, the following parameters should be used:

```
NFASGroupNumber 0 = 1
NFASGroupNumber 1 = 1
NFASGroupNumber 2 = 1
NFASGroupNumber 3 = 1
DchConfig 0 = 0    ;Primary T1 trunk
DchConfig 1 = 1    ;Backup T1 trunk
DchConfig 2 = 2    ;B-Channel NFAS trunk
DchConfig_3 = 2    ;B-channel NFAS trunk
```

If there is no NFAS Backup trunk, the following configuration should be used:

```
ISDNNFASInterfaceID_0 = 0
ISDNNFASInterfaceID_1 = 2
ISDNNFASInterfaceID_2 = 3
ISDNNFASInterfaceID_3 = 4
ISDNIBehavior = 512   ;This parameter should be added because of
;ISDNNFASInterfaceID coniguration above
NFASGroupNumber_0 = 1
NFASGroupNumber_1 = 1
NFASGroupNumber_2 = 1
NFASGroupNumber_3 = 1
DchConfig_0 = 0    ;Primary T1 trunk
DchConfig_1 = 2    ;B-Channel NFAS trunk
DchConfig_2 = 2    ;B-Channel NFAS trunk
DchConfig_3 = 2    ;B-channel NFAS trunk
```

11.4.3 Creating an NFAS-Related Trunk Configuration

The procedures for creating and deleting an NFAS group must be performed in the correct order, as described below.

➢ **To create an NFAS Group:**

1. If there's a backup ('secondary') trunk for this group, it must be configured first.
2. Configure the primary trunk before configuring any NFAS ('slave') trunk.
3. Configure NFAS ('slave') trunks.

➢ **To stop / delete an NFAS Group:**

1. Stop or delete (by setting ProtocolType to 0, i.e., 'None') all NFAS ('slave') trunks.
2. Stop or delete (by setting ProtocolType to 0, i.e., 'None') the backup trunk if a backup trunk exists.
3. Stop or delete (by setting ProtocolType to 0, i.e., 'None') the primary trunk.

Notes:

- All trunks in the group must be configured with the same values for trunk parameters TerminationSide, ProtocolType, FramingMethod, and LineCode.
- After stopping or deleting the backup trunk, delete the group and then reconfigure it.

11.5 Redirect Number and Calling Name (Display)

The following tables define the device's redirect number and calling name (Display) support for various ISDN variants according to NT (Network Termination) / TE (Termination Equipment) interface direction:

Table 11-3: Calling Name (Display)

NT/TE Interface	DMS-100	NI-2	4/5ESS	Euro ISDN	QSIG
NT-to-TE	Yes	Yes	No	Yes	Yes
TE-to-NT	Yes	Yes	No	No	Yes

Table 11-4: Redirect Number

NT/TE Interface	DMS-100	NI-2	4/5ESS	Euro ISDN	QSIG
NT-to-TE	Yes	Yes	Yes	Yes	Yes
TE-to-NT	Yes	Yes	Yes	Yes*	Yes

* When using ETSI DivertingLegInformation2 in a Facility IE (not Redirecting Number IE).

11.6 Automatic Gain Control (AGC)

Automatic Gain Control (AGC) adjusts the energy of the output signal to a required level (i.e., volume). This feature compensates for near-far gain differences. AGC estimates the energy of the incoming signal (from the IP or PSTN, determined by the parameter AGCRedirection), calculates the essential gain, and then performs amplification. Feedback ensures that the output signal is not clipped. You can define the required Gain Slope in decibels per second (using the parameter AGCGainSlop) and the required signal energy threshold (using the parameter AGCTargetEnergy).

When the AGC first detects an incoming signal, it begins operating in Fast Mode, which allows the AGC to adapt quickly when a conversation starts. This means that the Gain Slope is 8 dB/sec for the first 1.5 seconds. After this period, the Gain Slope is changed to the user-defined value. You can disable or enable the AGC's Fast Mode feature, using the *ini* file parameter AGCDisableFastAdaptation. After Fast Mode is used, the signal should be off for two minutes in order to have the feature turned on again.

To configure AGC, refer to "Configuring the IP Media Settings" on page 75.

12 Tunneling Applications

This section discusses the device's support for tunneling applications.

12.1 TDM Tunneling

The device's TDM Tunneling feature allows you to tunnel groups of digital trunk spans or timeslots (B-channels) over the IP network. TDM Tunneling utilizes the device's internal routing (without Proxy control) capabilities to receive voice and data streams from TDM (E1/T1/J1/) spans or individual timeslots, convert them into packets, and then transmit them over the IP network (using point-to-point or point-to-multipoint device distributions). A device opposite it (or several devices when point-to-multipoint distribution is used) converts the IP packets back into TDM traffic. Each timeslot can be targeted to any other timeslot within a trunk in the opposite device.

When TDM Tunneling is enabled (the parameter EnableTDMoverIP is set to '1') on the originating device, the originating device automatically initiates SIP calls from all enabled B-channels belonging to the E1/T1/J1 spans that are configured with the protocol type 'Transparent' (for ISDN trunks) or 'Raw CAS' (for CAS trunks). The called number of each call is the internal phone number of the B-channel from where the call originates. The 'Inbound IP Routing Table' is used to define the destination IP address of the terminating device. The terminating device automatically answers these calls if its E1/T1 protocol type is set to 'Transparent' (ProtocolType = 5) or 'Raw CAS' (ProtocolType = 3 for T1 and 9 for E1) and the parameter ChannelSelectMode is set to 0 (By Phone Number).

Note: It's possible to configure both devices to also operate in symmetric mode. To do so, set EnableTDMOverIP to 1 and configure the 'Outbound IP Routing Table' in both devices. In this mode, each device (after it's reset) initiates calls to the second device. The first call for each B-channel is answered by the second device.

The device continuously monitors the established connections. If for some reason, one or more calls are released, the device automatically re-establishes these 'broken' connections. In addition, when a failure in a physical trunk or in the IP network occurs, the device re-establishes the tunneling connections when the network is restored.

Note: It's recommended to use the keep-alive mechanism for each connection, by activating the 'session expires' timeout and using Re-INVITE messages.

The device supports the configuration (TDMoIPInitiateInviteTime and TDMoIPInviteRetryTime parameters) of the following timers for the TDM-over-IP tunneling application:

- Time between successive INVITEs sent from the same E1/T1 trunk.
- Time between call release and the new INVITE that is sent on the same channel. The call can be released if the device receives a 4xx or 5xx response.

By utilizing the 'Profiles' mechanism (refer to "Coders and Profile Definitions" on page 127), you can configure the TDM Tunneling feature to choose different settings based on a timeslot or groups of timeslots. For example, you can use low-bit-rate vocoders to transport voice and 'Transparent' coder to transport data (e.g., for D-channel). You can also use Profiles to assign ToS (for DiffServ) per source - a timeslot carrying data or signaling is assigned a higher priority value than a timeslot carrying voice.

For tunneling of E1/T1 CAS trunks, set the protocol type to 'Raw CAS' (ProtocolType = 3 / 9) and enable RFC 2833 CAS relay mode ('CAS Transport Type' parameter is set to 'CAS RFC2833 Relay').

Note: For TDM over IP, the parameter CallerIDTransportType must be set to '0' (disabled), i.e., transparent.

Below is an example of *ini* files for two devices implementing TDM Tunneling for four E1 spans. Note that in this example both devices are dedicated to TDM tunneling.

Terminating Side:

```
EnableTDMOverIP = 1

;E1 TRANSPARENT 31
ProtocolType_0 = 5
ProtocolType_1 = 5
ProtocolType_2 = 5
ProtocolType_3 = 5

[PREFIX]
FORMAT PREFIX_Index = PREFIX_DestinationPrefix,
PREFIX_DestAddress, PREFIX_SourcePrefix, PREFIX_ProfileId,
PREFIX_MeteringCode, PREFIX_DestPort;
Prefix_1 = '*,10.8.24.12';
[\PREFIX]

;IP address of the device in the opposite
;location

;Channel selection by Phone number.
ChannelSelectMode = 0

;Profiles can be used do define different coders per B-channels
;such as Transparent

;coder for B-channels (timeslot 16) that carries PRI ;signaling.
[TrunkGroup]
FORMAT TrunkGroup_Index = TrunkGroup_TrunkGroupNum,
TrunkGroup_FirstTrunkId, TrunkGroup_LastTrunkId,
TrunkGroup_FirstBChannel, TrunkGroup_LastBChannel,
TrunkGroup_FirstPhoneNumber, TrunkGroup_ProfileId,
TrunkGroup_Module;
TrunkGroup_1 = 0,0,0,1,31,1000,1;
TrunkGroup_1 = 0,1,1,1,31,2000,1;
TrunkGroup_1 = 0,2,2,1,31,3000,1;
TrunkGroup_1 = 0,3,3,1,31,4000,1;
TrunkGroup_1 = 0,0,0,16,16,7000,2;
TrunkGroup_1 = 0,1,1,16,16,7001,2;
TrunkGroup_1 = 0,2,2,16,16,7002,2;
TrunkGroup_1 = 0,3,3,16,16,7003,2;
[/TrunkGroup]

[ CodersGroup0 ]
FORMAT CodersGroup0_Index = CodersGroup0_Name, CodersGroup0_pTime,
CodersGroup0_rate, CodersGroup0_PayloadType, CodersGroup0_Sce;
CodersGroup0_0 = g7231;
CodersGroup0_1 = Transparent;
[ \CodersGroup0 ]
```

```
[TelProfile]
FORMAT TelProfile Index = TelProfile ProfileName,
TelProfile_TelPreference, TelProfile_CodersGroupID,
TelProfile IsFaxUsed, TelProfile JitterBufMinDelay,
TelProfile JitterBufOptFactor, TelProfile IPDiffServ,
TelProfile_SigIPDiffServ, TelProfile_DtmfVolume,
TelProfile InputGain, TelProfile VoiceVolume,
TelProfile EnableReversePolarity,
TelProfile_EnableCurrentDisconnect,
TelProfile EnableDigitDelivery, TelProfile EnableEC,
TelProfile MWIAnalog, TelProfile MWIDisplay,
TelProfile_FlashHookPeriod, TelProfile_EnableEarlyMedia,
TelProfile ProgressIndicator2IP;
TelProfile 1 = voice,$$,1,$$,$$,$$,$$,$$,$$,$$;
TelProfile 2 = data,$$,2,$$,$$,$$,$$,$$,$$,$$;
[\TelProfile]
```

Originating Side:

```
;E1 TRANSPARENT 31
ProtocolType 0 = 5
ProtocolType 1 = 5
ProtocolType 2 = 5
ProtocolType_3 = 5

;Channel selection by Phone number.
ChannelSelectMode = 0

[TrunkGroup]
FORMAT TrunkGroup Index = TrunkGroup TrunkGroupNum,
TrunkGroup FirstTrunkId, TrunkGroup LastTrunkId,
TrunkGroup FirstBChannel, TrunkGroup LastBChannel,
TrunkGroup_FirstPhoneNumber, TrunkGroup_ProfileId,
TrunkGroup Module;
TrunkGroup 0 = 0,0,0,1,31,1000,1;
TrunkGroup 0 = 0,1,1,1,31,2000,1;
TrunkGroup 0 = 0,2,2,1,31,3000,1;
TrunkGroup 0 = 0,3,1,31,4000,1;
TrunkGroup 0 = 0,0,0,16,16,7000,2;
TrunkGroup 0 = 0,1,1,16,16,7001,2;
TrunkGroup 0 = 0,2,2,16,16,7002,2;
TrunkGroup 0 = 0,3,3,16,16,7003,2;
[\TrunkGroup]

[ CodersGroup0 ]
FORMAT CodersGroup0 Index = CodersGroup0 Name, CodersGroup0 pTime,
CodersGroup0 rate, CodersGroup0 PayloadType, CodersGroup0 Sce;
CodersGroup0 0 = g7231;
CodersGroup0 1 = Transparent;
[ \CodersGroup0 ]

[TelProfile]
FORMAT TelProfile Index = TelProfile ProfileName,
TelProfile_TelPreference, TelProfile_CodersGroupID,
TelProfile IsFaxUsed, TelProfile JitterBufMinDelay,
TelProfile JitterBufOptFactor, TelProfile IPDiffServ,
TelProfile_SigIPDiffServ, TelProfile_DtmfVolume,
TelProfile InputGain, TelProfile VoiceVolume,
TelProfile EnableReversePolarity,
TelProfile_EnableCurrentDisconnect,
TelProfile EnableDigitDelivery, TelProfile EnableEC,
TelProfile MWIAnalog, TelProfile MWIDisplay,
TelProfile_FlashHookPeriod, TelProfile_EnableEarlyMedia,
TelProfile ProgressIndicator2IP;
TelProfile 1 = voice,$$,1,$$,$$,$$,$$,$$,$$,$$
TelProfile 2 = data,$$,2,$$,$$,$$,$$,$$,$$,$$
[\TelProfile]
```

12.1.1 DSP Pattern Detector

For TDM tunneling applications, you can use the DSP pattern detector feature to initiate the echo canceller at call start. The device can be configured to support detection of a specific one-byte idle data pattern transmitted over digital E1/T1 timeslots. The device can be configured to detect up to four different one-byte data patterns. When the defined idle data pattern is detected, the channel resets its echo canceller.

The following parameters must be configured:

- EnableDSPIPMDetectors = 1
- EnablePatternDetector = 1
- Pattern Detector Threshold: PDThreshold - defines the number of consecutive patterns to trigger the pattern detection event. For example: PDThreshold = 5

Detection Pattern: PDPattern - defines the patterns that can be detected by the Pattern Detector. For example: PDPattern = 84, 85, 212, 213 (for idle patterns: 54, 55, D4 and D5)

12.2 QSIG Tunneling

The device supports QSIG tunneling over SIP according to IETF draft 'Tunnelling of QSIG over SIP' (draft-elwell-sipping-qsig-tunnel-03) and the ECMA-355/ISO/IEC 22535 standard. This method enables all QSIG messages to be sent as raw data in corresponding SIP messages using a dedicated message body. This mechanism is useful for two QSIG subscribers (connected to the same or different QSIG PBX) to communicate with each other over an IP network. Tunneling is supported in both directions (Tel-to-IP and IP-to-Tel).

The term tunneling means that messages are transferred 'as is' to the remote side without being converted (QSIG→SIP→QSIG). The advantage of tunneling over QSIG-to-SIP interworking is that by using interworking, QSIG functionality can only be partially achieved. When tunneling is used, all QSIG capabilities are supported, whereas the tunneling medium (the SIP network) does not need to process these messages.

QSIG messages are transferred in SIP messages in a separate Multipurpose Internet Mail Extensions (MIME) body. Therefore, if a message contains more than one body (e.g., SDP and QSIG), multipart MIME must be used. The Content-Type of the QSIG tunneled message is 'application/QSIG'. In addition, the device adds a Content-Disposition header in the following format:

```
Content-Disposition: signal; handling=required.
```

- **Call setup (originating device):** The QSIG SETUP request is encapsulated in the SIP INVITE message without being altered. After the SIP INVITE request is sent, the device doesn't encapsulate the subsequent QSIG message until a SIP 200 OK response is received. If the originating device receives a 4xx, 5xx, or 6xx response, it disconnects the QSIG call with a 'no route to destination' cause.
- **Call setup (terminating device):** After the terminating device receives a SIP INVITE request with a 'Content-Type: application/QSIG', it sends the encapsulated QSIG SETUP message to the Tel side and sends a 200 OK response (no 1xx response is sent) to IP. The 200 OK response includes an encapsulated QSIG CALL PROCEEDING message (without waiting for a CALL PROCEEDING message from the Tel side). If tunneling is disabled and the incoming INVITE includes a QSIG body, a 415 response is sent.

- **Mid-call communication:** After the SIP connection is established, all QSIG messages are encapsulated in SIP INFO messages.
- **Call tear-down:** The SIP connection is terminated once the QSIG call is complete. The RELEASE COMPLETE message is encapsulated in the SIP BYE message that terminates the session.

To enable QSIG tunneling, set the parameter EnableQSIGTunneling to 1 on both the originating and terminating devices, and the parameter ISDNDuplicateQ931BuffMode to 128 (duplicate all messages) (both parameters are described in "ISDN and CAS Interworking Parameters" on page 392).

Reader's Notes

13 IP Media Capabilities

This section provides information on the device's media server capabilities:

- Multi-party conferencing (refer to "Conference Server" on page 641)
- Playing and recording Announcements (refer to "Announcement Server" on page 656)
- IP-to-IP Transcoding (refer to "IP-to-IP Transcoding" on page 528)
- Voice XML Interpreter (refer to Voice XML Interpreter on page 681)

Note: This section is applicable only to Mediant 1000.

The device conference, transcoding, announcement and media server applications can be used separately, each on a different platform, or all on the same device. The SIP URI name in the INVITE message is used to identify the resource (media server, conference or announcement) to which the SIP session is addressed.

The number of DSP channels that are allocated for IP conferences, transcoding and IP announcements is determined by the parameter MediaChannels. Other DSP channels can be used for PSTN media server.

The device's SIP implementation is based on the decomposition model described in the following IETF drafts:

- 'A Multi-party Application Framework for SIP' (draft-ietf-sipping-cc-framework-06.txt)
- 'Models for Multi Party Conferencing in SIP' (draft-ietf-sipping-conferencing-framework-05.txt)
- 'A Framework for Conferencing with the Session Initiation Protocol (SIP)' (RFC 4353)
- 'Basic Network Media Services with SIP' (RFC 4240)
- 'Media Server Control Markup Language (MSCML) and Protocol' (draft-vandyke-mscml-06.txt)

Note: To use the device's advanced Announcement capabilities, it's essential that the *ini* file parameter AMSProfile be set to 1.

13.1 Conference Server

The device supports dial-in, multi-party conferencing. In conference applications, the device functions as a centralized conference bridge. In ad-hoc or prearranged conferences, users 'invite' the conference bridge. The conference bridge mixes the media and sends it to all participants.

The device supports the following interfaces for conferencing:

- Simple, according to NetAnn (refer to "Simple Conferencing (NetAnn)" on page 642)
- Advanced, according to MSCML (refer to "Advanced Conferencing (MSCML)" on page 644)

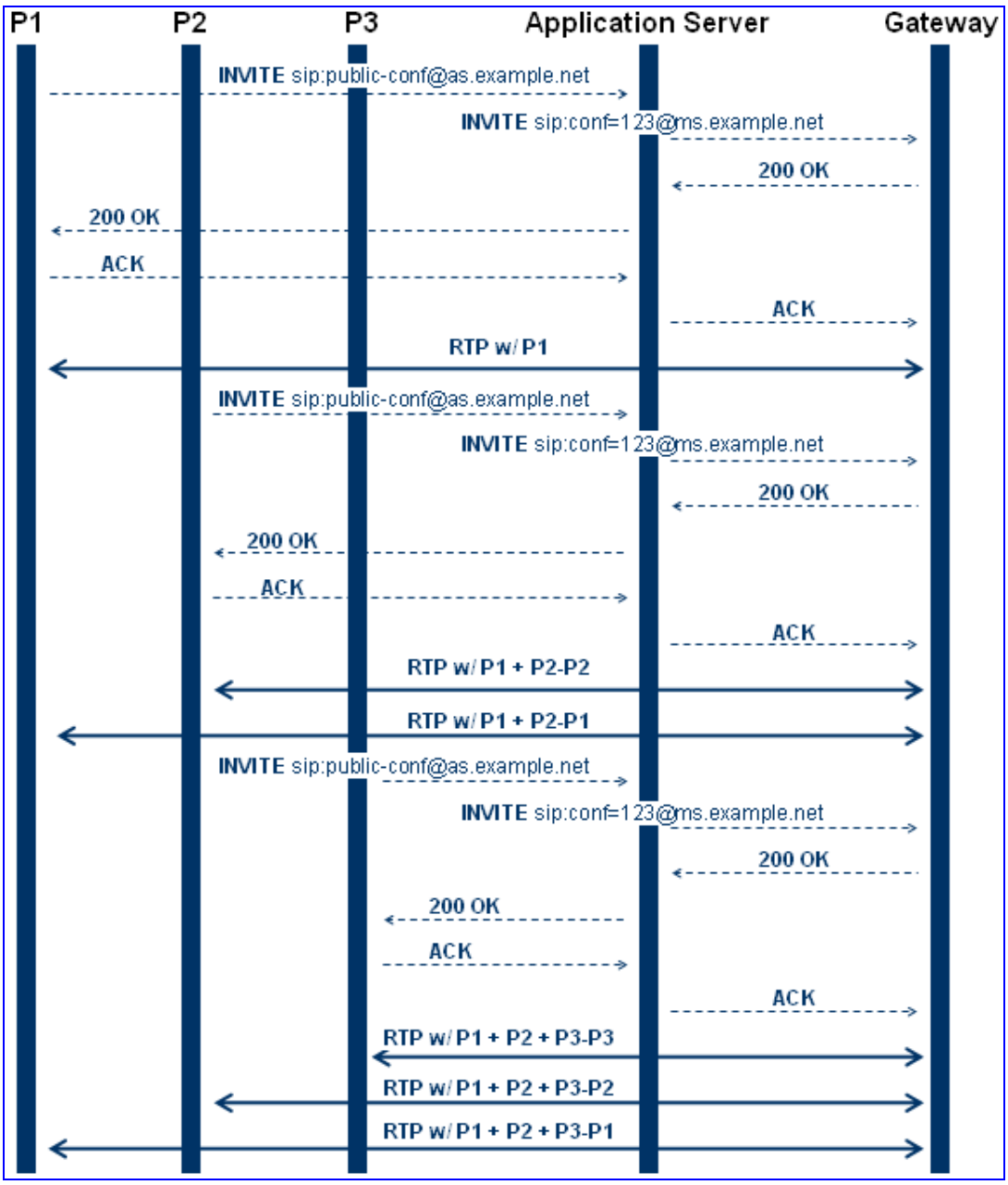
Note: The conference application is a special order option.

13.1.1 Simple Conferencing (NetAnn)

13.1.1.1 SIP Call Flow

A SIP call flow for simple conferencing is shown below:

Table 13-1: Simple Conferencing SIP Call Flow



13.1.1.2 Creating a Conference

The device creates a conference call when the first user joins the conference. To create a conference, the Application Server sends a regular SIP INVITE message to the device. The User Part of that Request-URI includes both the Conference Service Identifier (indicating that the requested Media Service is a Conference) and a Unique Conference Identifier (identifying a specific instance of a conference).

```
INVITE sip: conf100@audiocodes.com SIP/2.0
```

By default, a request to create a conference reserves three resources on the device. It is possible to reserve a larger number of resources in advance by adding the number of required participants to the User Part of the Request-URI. For example, '6conf100' reserves six resources for the duration of the conference. If the device can allocate the requested number of resources, it responds with a 200 OK.

The Conference Service Identifier can be set using the *ini* file parameter ConferenceID. By default, it is set to 'conf'.

13.1.1.3 Joining a Conference

To join an existing conference, the Application Server sends a SIP INVITE message with the same Request-URI as the one that created the conference. Each conference participant can use a different coder negotiated with the device using usual SIP negotiation.

If more than the initially requested number of participants try to join the conference (i.e., four resources were reserved and a fifth INVITE is received) and the device has an available resource, the request is granted.

If an INVITE to join an existing conference is received with a request to reserve a larger number of participants than initially requested, it is granted if the device has available resources. A request for a smaller number of participants is not granted as this may create a situation where existing legs would need to be disconnected.

The maximal number of participants in a single conference is 60. The maximal number of participants that actually participate in the mix at a given time is three (the loudest legs).

The Application Server can place a participant on Hold/Un-hold by sending the appropriate SIP Re-INVITE on that participant dialog.

13.1.1.4 Terminating a Conference

The device never disconnects an existing conference leg. If a BYE is received on an existing leg, it is disconnected, but the resource is still saved if the same leg (or a different one) wants to re-join the conference. This logic occurs only for the initial number of reserved legs.

For example:

1. INVITE reserves three legs.
2. A, B, and C join the conference.
3. A disconnects.
4. A joins (guaranteed).
5. D joins.

6. A disconnects.
7. A joins (not guaranteed).

Sending a BYE request to the device terminates the participant's SIP session and removes it from the conference. The final BYE from the last participant ends the conference and releases all conference resources.

13.1.1.5 PSTN Participants

Adding PSTN participants is done by performing a loopback from the IP side (the device's IP address is configured in the 'Outbound IP Routing Table'). If the destination phone number in the incoming call from the PSTN is equal to the Conference Service Identifier and Unique Conference Identifier, the participant joins the conference.

A PSTN participant uses two DSP channels (caused by the IP loopback).

13.1.2 Advanced Conferencing (MSCML)

13.1.2.1 Creating a Conference

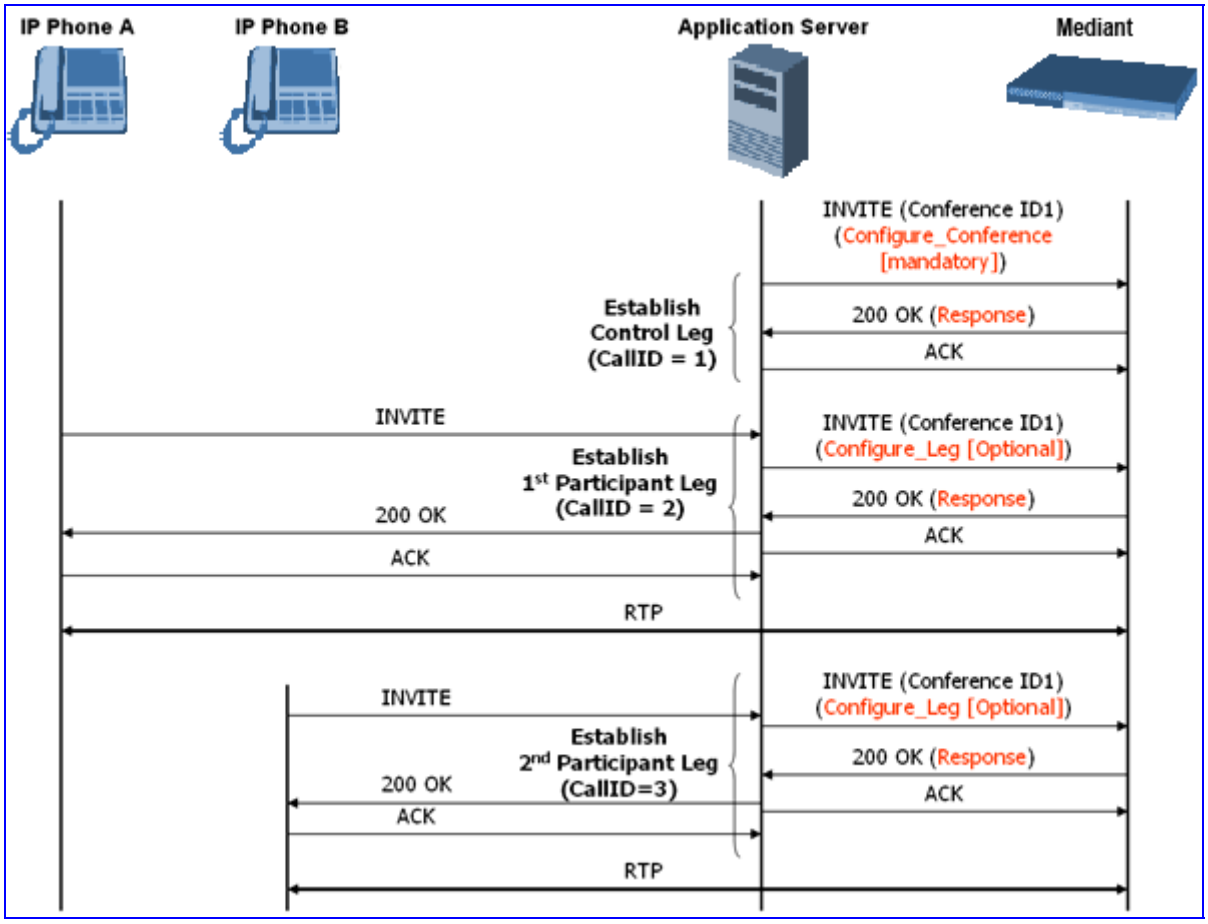
The device creates a conference call when the first INVITE is received from the Application Server (same as NetAnn). The Unique Conference Identifier is used to join participants to the same conference. This first INVITE must include a <configure_conference> MSCML request body. If this body is not included, a simple conference is established. This first leg is the Control Leg, which is different from a regular Participant Leg. The Control Leg is used to perform operations for the whole conference.

The MSCML response to the first INVITE is sent in the 200 OK SIP response. If no error occurs, the response is:
<response request="configure_conference" code="200" text="OK"/>.

The <configure_conference> can include the following attributes:

- **Id:** identification number of the MSCML request. This is used to correlate between MSCML requests and responses.
- **Reservedtalkers:** defines the maximum number of talker legs. As the device does not support “listener only” legs, this actually sets the maximum number of participants in the conference. The device reserves this number of participants for the entire duration of the conference. If a participant leg decides to leave the conference by issuing a BYE, the resource is not freed, thereby allowing that same leg (or a new one) to join at any stage.
- **Reserveconfmedia:** determines if Media Services such as Play or Record can be applied to the conference. If set to Yes, the device reserves the necessary amount of resources to play an announcement to the whole conference or record the whole conference. The Application Server can change the value of reserveconfmedia during an existing conference. By default, reserveconfmedia is set to Yes.

Table 13-2: Advanced Conferencing SIP Call Flow



13.1.2.2 Joining a Conference

To join an existing conference, the Application Server sends a SIP INVITE message with the same Request-URI as the one that created the conference. The INVITE message may include a <configure_leg> MSCML request body. If not included, defaults are used for that leg attributes.

The <configure_leg> can include the following attributes:

- **Id:** identification number of the MSCML request. This is used to correlate between MSCML requests and responses.
- **Type:** Talker / Listener. If set to Listener, the incoming RTP from that leg does not participate in the conference mix. The default is Talker.
- **Mixmode:**
  - Full: RTP from this leg participates in the mix (default).
  - Mute: RTP from this leg is not participating in the mix.
  - Private: RTP from this leg can only hear participants within a conference team (<teammate>) to which it belongs (see below).

The <configure_team> element enables clients to create personalized mixes for scenarios where the standard mixmode settings do not provide sufficient control. The <configure_team> element is a child of <configure_leg>. The <configure_team> element, containing one or more <teammate> elements, specifies those participants that should be present in this participant's personalized mix. The <configure_team> element supports several commands: set, add, remove, and query.

The participants are identified in the <teammate> elements by their IDs that are assigned in their <configure_leg> element. The team configuration is implicitly symmetric, i.e. if participant A defines participant B as its team member, implicitly participant B defines participant A as its team member.

A "coaching" example:

Table 13-3: MSCML Conferencing with Personalized Mixes

Participant	ID	Team Members	Mixmode	Hears
Supervisor	"supervisor"	Agent	Private	Customer and Agent
Agent	"agent"	Supervisor	Full	Customer and Supervisor
Customer	"customer"		Full	Agent

This scenario is established as follows:

1. Conference is created on the control leg with <configure_conference>.
2. Coach leg joins and issues:

```
<configure_leg id="supervisor" mixmode="private"/>
```

3. Agent leg joins and issues:

```
<configure_leg id="agent">
         <configure_team action="set">
           <teammate id="supervisor"/>
         </configure_team>
       </configure_leg>
```

4. Customer joins and issues:

```
<configure_leg id="customer"/>
```

13.1.2.3 Modifying a Conference

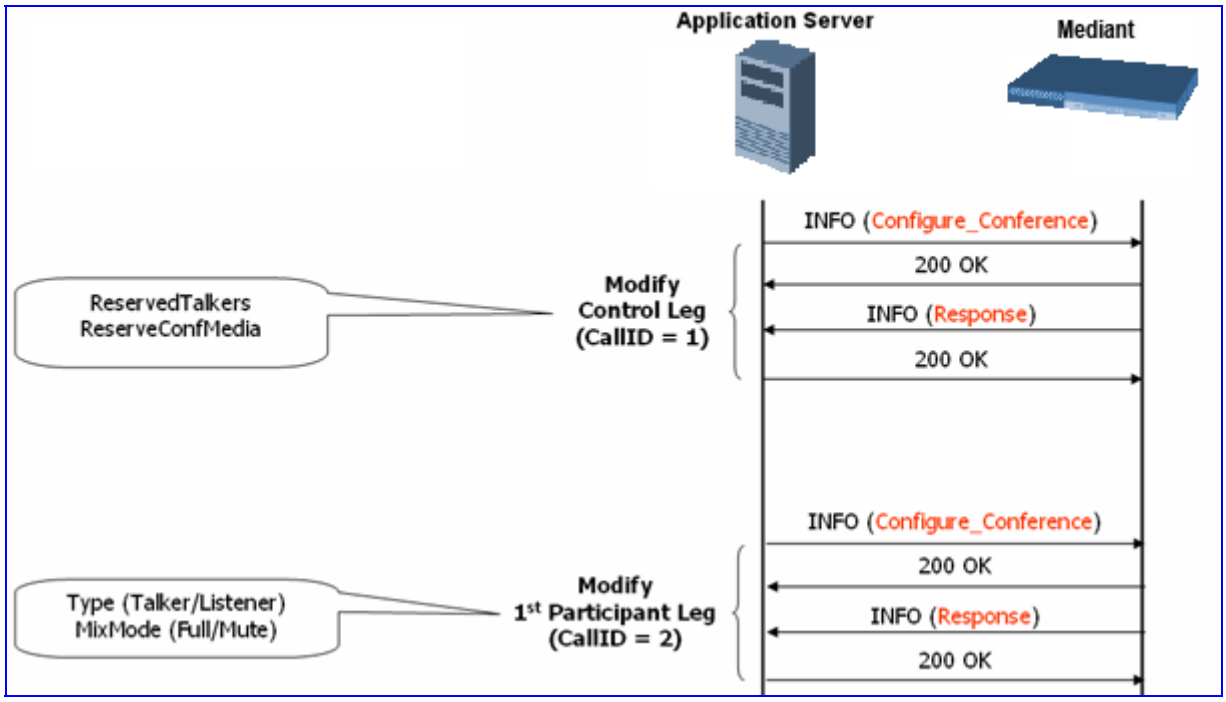
To modify an existing conference, INFO messages are used. Each INFO message carries an MSCML request. The MSCML response is included in an INFO message back from the device to the Application Server. It is possible to modify an entire conference (by issuing requests on the Control Leg) or only a certain participant (by issuing requests on that specific leg).

To modify the entire conference, a <configure_conference> MSCML request body is sent in an INFO message on the Control Leg SIP dialog. Using this request, the Application Server can modify the following attributes:

- **Reservedtalkers:** If the Application Server sets a number that is lower than the initial number requested in the INVITE, then the request is not granted. If the number is higher than the initial number, the device sends a success response in the response INFO.
- **Reserveconfmedia:** If the necessary resources for applying Media Services on the entire conference were reserved in advance, then by setting reserveconfmedia to Yes, it is reserved. If set to No, the device can free the resource.

To modify a certain Participant Leg, a <configure_leg> MSCML request body is sent in an INFO message on that leg SIP dialog. Using this request, the Application Server can modify any of the attributes defined for the <configure_leg> request.

Table 13-4: Modifying a Conference - SIP Call Flow

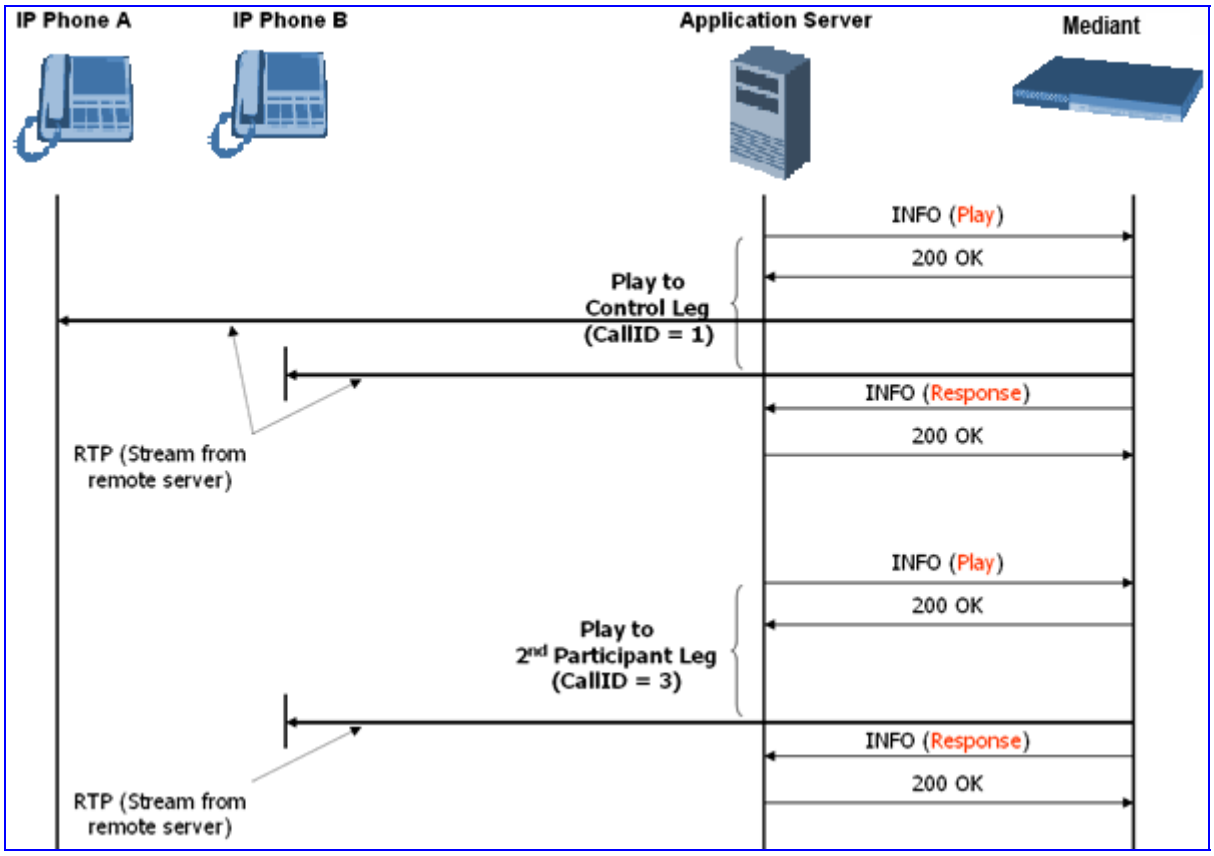


13.1.2.4 Applying Media Services on a Conference

The Application Server can issue a Media Service request (<play>, <playcollect>, or <playrecord>) on either the Control Leg or a specific Participant Leg. For a Participant Leg, all three requests are applicable. For the Control Leg, the <playcollect> is not applicable as there is no way to collect digits from the whole conference.

When issuing a Media Service on the Control Leg, it affects all Participant Legs in the conference, e.g., play an announcement. When issuing a Media Service on a Participant Leg, it affects the specific leg only.

Table 13-5: Applying Media Services on a Conference - SIP Call Flow



13.1.2.5 Active Speaker Notification

After an advanced conference is established, the Application Server can subscribe to the device to receive notifications of the current set of active speakers in a conference at any given moment. This feature is referred to as *Active Speaker Notification* (ASN) and is designed according to the MSCML standard. Notifications provide information on the number of active participants and their details.

The notifications are sent unsolicited at specific intervals requested by the application and only when a change in the number of active conference speakers occurs. If a change in the speakers list occurs, the server issues an INVITE to the specific SIP UA, and then transfers the call to the UA.

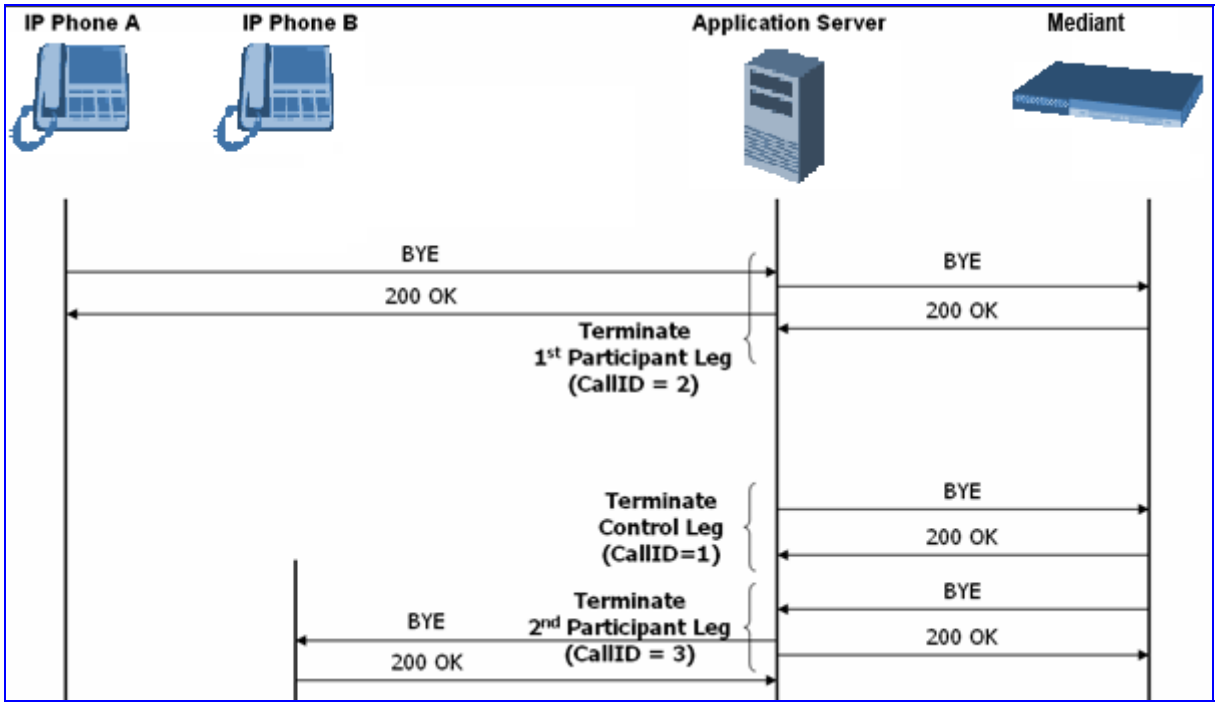
Event notifications are sent in SIP INFO messages, as shown in the example below of XML Response Generated for ASN:

```
<?xml version="1.0" encoding="utf-8"?>
<MediaServerControl version="1.0">
<notification>
<conference uniqueID="3331" numtalkers="1">
<activetalkers>
<talker callID="98142661715120001936l9@10.8.27.118"/>
</activetalkers>
</conference>
</notification>
</MediaServerControl>
```

13.1.2.6 Terminating a Conference

To remove a leg from a conference, the Application Server issues a SIP BYE request on the selected dialog representing the conference leg. The Application Server can terminate all legs in a conference by issuing a SIP BYE request on the Control Leg. If one or more participants are still in the conference when the device receives a SIP BYE request on the Control Leg, the device issues SIP BYE requests on all of the remaining conference legs to ensure a clean up of the legs.

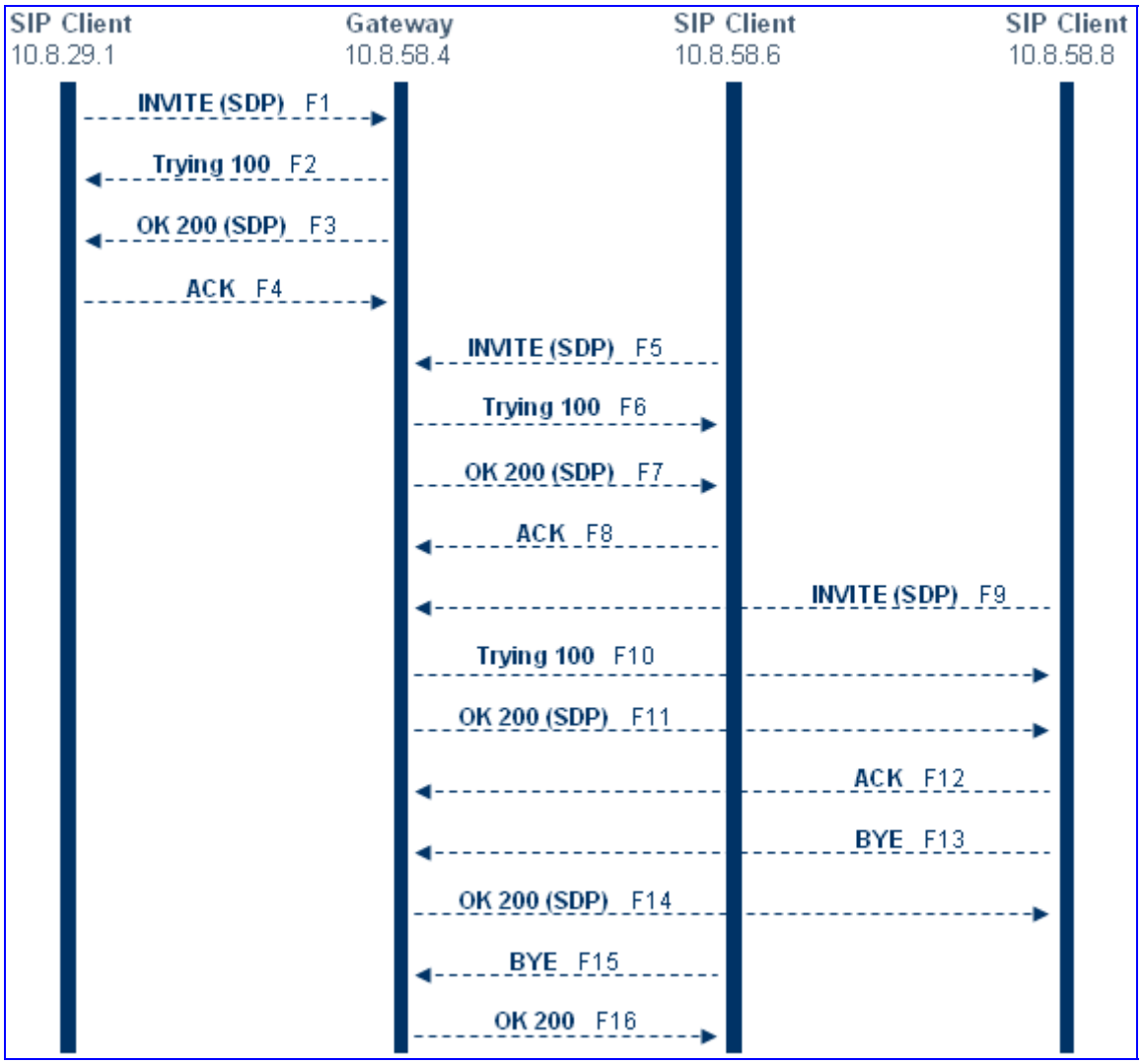
Table 13-6: Terminating a Conference - SIP Call Flow



13.1.3 Conference Call Flow Example

The call flow, shown in the following figure, describes SIP messages exchanged between the device (10.8.58.4) and three conference participants (10.8.29.1, 10.8.58.6 and 10.8.58.8).

Table 13-7: Conference Call Flow Example



1. **SIP MESSAGE 1:** 10.8.29.1:5060 -> 10.8.58.4:5060

```
INVITE sip:conf100@10.8.58.4;user=phone SIP/2.0
Via: SIP/2.0/UDP 10.8.29.1;branch=z9hG4bKacRHmJhMj
Max-Forwards: 70
From: <sip:100@10.8.8.10>;tag=1c352329022
To: <sip:conf100@10.8.58.4;user=phone>
Call-ID: 1792526528qlax@10.8.29.1
CSeq: 1 INVITE
Contact: <sip:100@10.8.29.1>
Supported: em,100rel,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
User-Agent: Audiocodes-Sip-Gateway-Mediant 1000/v.5.40A.010.006
Content-Type: application/sdp
Content-Length: 216
```

```
v=0
o=AudiocodesGW 663410 588654 IN IP4 10.8.29.1
s=Phone-Call
c=IN IP4 10.8.29.1
t=0 0
m=audio 6000 RTP/AVP 8 96
a=rtpmap:8 pcma/8000
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
a=ptime:20
a=sendrecv
```

2. **SIP MESSAGE 2:** 10.8.58.4:5060() -> 10.8.29.1:5060()

```
SIP/2.0 100 Trying
Via: SIP/2.0/UDP 10.8.29.1;branch=z9hG4bKacRHmJhMj
From: <sip:100@10.8.8.10>;tag=1c352329022
To: <sip:conf100@10.8.58.4;user=phone>;tag=1c222574568
Call-ID: 1792526528qlax@10.8.29.1
CSeq: 1 INVITE
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
Server: Audiocodes-Sip-Gateway-Mediant 1000/v.5.40A.010.006
Content-Length: 0
```

3. **SIP MESSAGE 3:** 10.8.58.4:5060 -> 10.8.29.1:5060

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.8.29.1;branch=z9hG4bKacRHmJhMj
From: <sip:100@10.8.8.10>;tag=1c352329022
To: <sip:conf100@10.8.58.4;user=phone>;tag=1c222574568
Call-ID: 1792526528qlax@10.8.29.1
CSeq: 1 INVITE
Contact: <sip:10.8.58.4>
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
Server: Audiocodes-Sip-Gateway-Mediant 1000/v.5.40A.010.006
Content-Type: application/sdp
Content-Length: 216

v=0
o=AudiocodesGW 820775 130089 IN IP4 10.8.58.4
s=Phone-Call
c=IN IP4 10.8.58.4
t=0 0
m=audio 7160 RTP/AVP 8 96
a=rtpmap:8 pcma/8000
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
a=ptime:20
a=sendrecv
```

4. SIP MESSAGE 4: 10.8.29.1:5060 -> 10.8.58.4:5060

```
ACK sip:10.8.58.4 SIP/2.0
Via: SIP/2.0/UDP 10.8.29.1;branch=z9hG4bKacbUrWtRo
Max-Forwards: 70
From: <sip:100@10.8.8.10>;tag=1c352329022
To: <sip:conf100@10.8.58.4;user=phone>;tag=1c222574568
Call-ID: 1792526528qlax@10.8.29.1
CSeq: 1 ACK
Contact: <sip:100@10.8.29.1>
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
User-Agent: Audiocodes-Sip-Gateway-Mediant 1000/v.5.40A.010.006
Content-Length: 0
```

5. SIP MESSAGE 5: 10.8.58.6:5060 -> 10.8.58.4:5060

```
INVITE sip:conf100@10.8.58.4;user=phone SIP/2.0
Via: SIP/2.0/UDP 10.8.58.6;branch=z9hG4bKacfowEuut
Max-Forwards: 70
From: <sip:600@10.8.8.10>;tag=1c201038291
To: <sip:conf100@10.8.58.4;user=phone>
Call-ID: 1008914574iYgW@10.8.58.6
CSeq: 1 INVITE
Contact: <sip:600@10.8.58.6>
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
User-Agent: Audiocodes-Sip-Gateway-Mediant 1000/v.4.60A.005.009
Content-Type: application/sdp
Content-Length: 313

v=0
o=AudiocodesGW 702680 202680 IN IP4 10.8.58.6
s=Phone-Call
c=IN IP4 10.8.58.6
t=0 0
m=audio 6000 RTP/AVP 4 8 0 110 96
a=rtpmap:4 g723/8000
a=fmtp:4 annexa=no
a=rtpmap:8 pcma/8000
a=rtpmap:0 pcmu/8000
a=rtpmap:110 AMR/8000/1
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
a=ptime:30
a=sendrecv
```

6. SIP MESSAGE 6: 10.8.58.4:5060 -> 10.8.58.6:5060

```
SIP/2.0 100 Trying
Via: SIP/2.0/UDP 10.8.58.6;branch=z9hG4bKacfowEuut
From: <sip:600@10.8.8.10>;tag=1c201038291
To: <sip:conf100@10.8.58.4;user=phone>;tag=1c1673415884
Call-ID: 1008914574iYgW@10.8.58.6
CSeq: 1 INVITE
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
Server: Audiocodes-Sip-Gateway-Mediant 1000/v.5.40A.010.006
Content-Length: 0
```

7. SIP MESSAGE 7: 10.8.58.4:5060 -> 10.8.58.6:5060

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.8.58.6;branch=z9hG4bKacfowEuut
From: <sip:600@10.8.8.10>;tag=1c201038291
To: <sip:conf100@10.8.58.4;user=phone>;tag=1c1673415884
Call-ID: 1008914574iYgW@10.8.58.6
CSeq: 1 INVITE Contact: <sip:conf100@10.8.58.4>
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
Server: Audiocodes-Sip-Gateway-Mediant 1000/v.5.40A.010.006
Content-Type: application/sdp
Content-Length: 236

v=0 o=AudiocodesGW 886442 597756 IN IP4 10.8.58.4
s=Phone-Call
c=IN IP4 10.8.58.4
t=0 0
m=audio 7150 RTP/AVP 4 96
a=rtpmap:4 g723/8000
a=fmtp:4 annexa=no
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
a=ptime:30
a=sendrecv
```

8. SIP MESSAGE 8: 10.8.58.6:5060 -> 10.8.58.4:5060

```
ACK sip:conf100@10.8.58.4 SIP/2.0
Via: SIP/2.0/UDP 10.8.58.6;branch=z9hG4bKacRRRZPXN
Max-Forwards: 70
From: <sip:600@10.8.8.10>;tag=1c201038291
To: <sip:conf100@10.8.58.4;user=phone>;tag=1c1673415884
Call-ID: 1008914574iYgW@10.8.58.6
CSeq: 1 ACK
Contact: <sip:600@10.8.58.6>
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
User-Agent: Audiocodes-Sip-Gateway-Mediant 1000/v.4.60A.005.009
Content-Length: 0
```

9. SIP MESSAGE 9: 10.8.58.8:5060 -> 10.8.58.4:5060

```
INVITE sip:conf100@10.8.58.4;user=phone SIP/2.0
Via: SIP/2.0/UDP 10.8.58.8;branch=z9hG4bKaczJpxnnv
Max-Forwards: 70
From: <sip:800@10.8.58.8>;tag=1c2419012378
To: <sip:conf100@10.8.58.4;user=phone>
Call-ID: 150852731NDDC@10.8.58.8
CSeq: 1 INVITE
Contact: <sip:800@10.8.58.8>
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
User-Agent: Audiocodes-Sip-Gateway-Mediant 1000/v.4.60A.005.009
Content-Type: application/sdp Content-Length: 236

v=0
o=AudiocodesGW 558246 666026 IN IP4 10.8.58.8
s=Phone-Call
c=IN IP4 10.8.58.8
```

```
t=0 0 m=audio 6000 RTP/AVP 4 96
a=rtpmap:4 g723/8000
a=fmtp:4 annexa=no
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
a=ptime:30
a=sendrecv
```

10. SIP MESSAGE 10: 10.8.58.4:5060 -> 10.8.58.8:5060

```
SIP/2.0 100 Trying
Via: SIP/2.0/UDP 10.8.58.8;branch=z9hG4bKaczJpxnnv
From: <sip:800@10.8.58.8>;tag=1c2419012378
To: <sip:conf100@10.8.58.4;user=phone>;tag=1c3203015250
Call-ID: 150852731NDDC@10.8.58.8
CSeq: 1 INVITE
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
Server: Audiocodes-Sip-Gateway-Mediant 1000/v.5.40A.010.006
Content-Length: 0
```

11. SIP MESSAGE 11: 10.8.58.4:5060 -> 10.8.58.8:5060

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.8.58.8;branch=z9hG4bKaczJpxnnv
From: <sip:800@10.8.58.8>;tag=1c2419012378
To: <sip:conf100@10.8.58.4;user=phone>;tag=1c3203015250
Call-ID: 150852731NDDC@10.8.58.8
CSeq: 1 INVITE
Contact: <sip:conf100@10.8.58.4>
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
Server: Audiocodes-Sip-Gateway-Mediant 1000/v.5.40A.010.006
Content-Type: application/sdp
Content-Length: 236

v=0
o=AudiocodesGW 385533 708665 IN IP4 10.8.58.4
s=Phone-Call
c=IN IP4 10.8.58.4
t=0 0
m=audio 7140 RTP/AVP 4 96
a=rtpmap:4 g723/8000
a=fmtp:4 annexa=no
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
a=ptime:30
a=sendrecv
```

12. SIP MESSAGE 12: 10.8.58.8:5060 -> 10.8.58.4:5060

```
ACK sip:conf100@10.8.58.4 SIP/2.0
Via: SIP/2.0/UDP 10.8.58.8;branch=z9hG4bKacisqqyow
Max-Forwards: 70
From: <sip:800@10.8.58.8>;tag=1c2419012378
To: <sip:conf100@10.8.58.4;user=phone>;tag=1c3203015250
Call-ID: 150852731NDDC@10.8.58.8
CSeq: 1 ACK
Contact: <sip:800@10.8.58.8>
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
User-Agent: Audiocodes-Sip-Gateway-Mediant 1000/v.5.40A.010.006
Content-Length: 0
```

13. SIP MESSAGE 13: 10.8.58.8:5060 -> 10.8.58.4:5060

```
BYE sip:conf100@10.8.58.4 SIP/2.0
Via: SIP/2.0/UDP 10.8.58.8;branch=z9hG4bKackSIyGww
Max-Forwards: 70
From: <sip:800@10.8.58.8>;tag=1c2419012378
To: <sip:conf100@10.8.58.4;user=phone>;tag=1c3203015250
Call-ID: 150852731NDDC@10.8.58.8
CSeq: 2 BYE
Contact: <sip:800@10.8.58.8>
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
User-Agent: Audiocodes-Sip-Gateway-Mediant 1000/v.5.40A.010.006
Content-Length: 0
```

14. SIP MESSAGE 14: 10.8.58.4:5060 -> 10.8.58.8:5060

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.8.58.8;branch=z9hG4bKackSIyGww
From: <sip:800@10.8.58.8>;tag=1c2419012378
To: <sip:conf100@10.8.58.4;user=phone>;tag=1c3203015250
Call-ID: 150852731NDDC@10.8.58.8
CSeq: 2 BYE
Contact: <sip:conf100@10.8.58.4>
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
Server: Audiocodes-Sip-Gateway-Mediant 1000/v.5.40A.010.006
Content-Length: 0
```

15. SIP MESSAGE 15: 10.8.58.6:5060 -> 10.8.58.4:5060

```
BYE sip:conf100@10.8.58.4 SIP/2.0
Via: SIP/2.0/UDP 10.8.58.6;branch=z9hG4bKacQypxnvl
Max-Forwards: 70
From: <sip:600@10.8.8.10>;tag=1c201038291
To: <sip:conf100@10.8.58.4;user=phone>;tag=1c1673415884
Call-ID: 1008914574iYgW@10.8.58.6
CSeq: 2 BYE
Contact: <sip:600@10.8.58.6>
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
User-Agent: Audiocodes-Sip-Gateway-Mediant 1000/v.5.40A.010.006
Content-Length: 0
```

16. SIP MESSAGE 16: 10.8.58.4:5060 -> 10.8.58.6:5060

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.8.58.6;branch=z9hG4bKacQypxnvl
From: <sip:600@10.8.8.10>;tag=1c201038291
To: <sip:conf100@10.8.58.4;user=phone>;tag=1c1673415884
Call-ID: 1008914574iYgW@10.8.58.6
CSeq: 2 BYE
Contact: <sip:conf100@10.8.58.4>
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
Server: Audiocodes-Sip-Gateway-Mediant 1000/v.5.40A.010.006
Content-Length: 0
```

13.2 Announcement Server

The device supports playing and recording of announcements (local Voice Prompts or HTTP streaming) and playing of Call Progress Tones over the IP network. Three different methods are available for playing and recording announcements:

- NetAnn for playing a single announcement (refer to "NetAnn Interface" on page 656)
- MSCML for playing single or multiple announcements and collecting digits (refer to "MSCML Interface" on page 657)

13.2.1 NetAnn Interface

The device supports playing announcements using NetAnn format (according to RFC 4240).

13.2.1.1 Playing a Local Voice Prompt

To play a single local Voice Prompt, the Application Server (or any SIP user agent) sends a regular SIP INVITE message with SIP URI that includes the NetAnn Announcement Identifier name. For example:

```
INVITE sip:annc@audiocodes.com; play=file://12 SIP/2.0
```

The left part of the SIP URI includes the string 'annc'. In the example above, the device starts playing announcement number 12 from the internal Voice Prompts file (file:// and http://localhost formats are supported). The NetAnn Announcement Identifier string is configured using the *ini* file (parameter NetAnnAnncID) or Web interface (refer to "Configuring the IPmedia Parameters" on page 179). Sending a BYE request terminates the SIP session and stops the playing of the announcement. If the played Voice Prompt reaches its end, the device initiates a BYE message to notify the Application Server that the session has ended.

13.2.1.2 Playing using HTTP/NFS Streaming

To play a single announcement via HTTP or NFS streaming, the Application Server (or any SIP user agent) sends a regular SIP INVITE message with SIP URI that includes the NetAnn Announcement Identifier name. For example:

```
INVITE sip:annc@ac.com;
play=http://server.net/gem/Hello.wav   SIP/2.0
```

The left part of the SIP URI includes the string 'annc' terminated by the IP address of the HTTP server, and the name and path of the file to be played. In the example above, the device starts playing the 'Hello.wav' file that resides in the folder 'server.net/gem'. The NetAnn Announcement Identifier string is configured using the *ini* file (parameter NetAnnAnncID) or Web interface (refer to "Configuring the IPmedia Parameters" on page 179). Sending a BYE request terminates the SIP session and stops the playing of the announcement. If the played announcement reaches its end, the device initiates a BYE message to notify the Application Server that the session is ended.

Notes:

- A 200 OK message is sent only after the HTTP connection is successfully established and the requested file is found. If the file isn't found, a 404 Not Found response is sent.
- To use NFS, the requested file system should be first mounted by using the NFS Servers table, see Configuring the NFS Settings on page 67.

13.2.1.3 Supported Attributes

When playing announcements, the following attributes are available:

- **Repeat:** defines the number of times the announcement is repeated. The default value is 1. The valid range is 1 to 1000, or -1 (i.e., repeats the message forever).
- **Delay:** defines the delay (in msec) between announcement repetitions. The default value is 0. The valid range is 1 to 3,600,000.
- **Duration:** defines the total duration (in msec) the announcement(s) are played. The default value is 0 (i.e., no limitation). The valid range is 1 to 3,600,000.

For example:

```
INVITE sip:annc@ac.com;
play=http://server.net/gem/Hello.wav; repeat=5;delay=10000 SIP/2.0
```

13.2.2 MSCML Interface

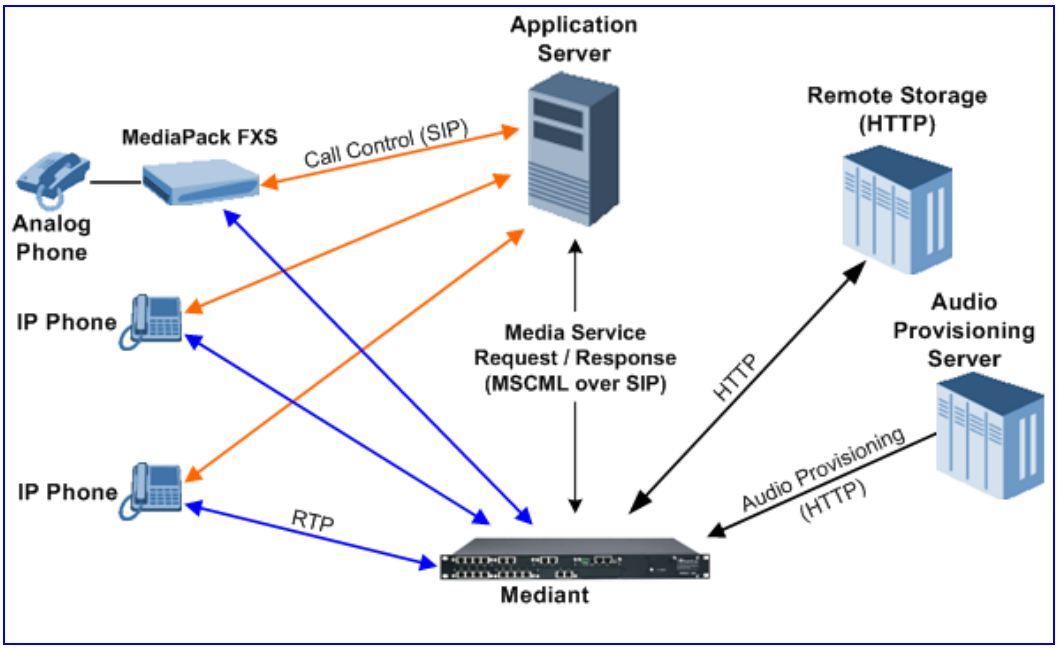
Media Server Control Markup Language (MSCML), according to IETF RFC 5022 is a protocol used in conjunction with SIP to provide advanced announcements handling. MSCML is implemented by adding an XML body to existing SIP INFO messages. Only a single message body (containing a single request or response) is allowed per message.

The device supports all the Interactive Voice Response (IVR) requirements for playing announcements, collecting digits, and recording (Play, PlayCollect, and PlayRecord).

Note: MSCML is only supported on devices operating with 128-MByte RAM size.

The following figure illustrates standard MSCML application architecture:

Table 13-8: MSCML Architecture



The architecture comprises the following components:

- **device:** Operating independently, the device controls and allocates its processing resources to match each application's requirements. Its primary role is to handle requests from the Application server for playing announcements and collecting digits.
- **Application Server:** An application platform that controls the call signaling. It interfaces with the device using MSCML. It instructs the media server to play announcements, collect digits and record voice streams.
- **Audio Provisioning Server (APS):** The APS is used for offline generation of *.dat files of audio packages including audio files, audio sequences, and different languages for variable announcement playing. These can be later loaded to the device.
- **Remote Storage:** An HTTP server that contains less-frequently used voice prompts for playback and to which voice stream recording is performed.
- **IP Phones / MediaPack:** Client applications.

13.2.2.1 Operation

The APS server can be used to generate two files - the audio package as a VP.dat file, and an XML file (segments.xml) that contains indices to the announcements stored on the VP.dat file for playing announcements. These two files can be loaded to the device using the Web interface.

An alternative method uses the AutoUpdate mechanism as described in the *Product Reference Manual*. Both the vp.dat and segments.xml files that were previously created using the APS should be located on an external storage server (HTTP, FTP). At startup, the device fetches the files from the remote storage. By using the AutoUpdate mechanism, the device periodically checks if new files are posted to the remote server and fetches these files.

The Application server communicates with the device using MSCML Requests (sent by the Application server), as shown in the example below:

```
<?xml version="1.0" encoding="utf-8"?>
  <MediaServerControl version="1.0">
    <request>
      ... request body ...
    </request>
  </MediaServerControl>
```

The device uses MSCML Responses (i.e., sent by the device) to reply to the Application server, as shown in the example below:

```
<?xml version="1.0" encoding="utf-8"?>
  <MediaServerControl version="1.0">
    <response>
      ... response body ...
    </response>
  </MediaServerControl>
```

To start an MSCML IVR call, the Application server (or any SIP user agent) sends a regular SIP INVITE message with a SIP URI that includes the MSCML Identifier name. For example:

```
INVITE sip:ivr@audiocodes.com SIP/2.0
```

The left part of the SIP URI includes the MSCML Identifier string 'ivr', which can be configured using the *ini* file (parameter MSCMLID) or Web interface (refer to "Configuring the IPmedia Parameters" on page 179).

After a call is established, SIP INFO messages are used to carry MSCML requests and responses. An INFO message that carries an MSCML body is identified by its content-type header that is set to 'application/mediaservercontrol+xml'.

Note that IVR requests are not queued. Therefore, if a request is received while another is in progress, the device stops the first operation and executes the new request. The device generates a response message for the first request and returns any data collected up to that point. If an application is required to stop a request in progress, it issues a <Stop> request. This request also causes the device to generate a response message.

The device supports basic IVR functions of playing announcements, collecting DTMF digits, and voice stream recording. These services are implemented using the following Request and Response messages:

- <Play> for playing announcements

- <PlayCollect> for playing announcements and collecting digits
- <PlayRecord> for playing announcements and recording voice
- <Stop> for stopping the playing of an announcement

The device sends a Response to each Request that is issued by the Application server.

The <Play>, <PlayCollect>, and <PlayRecord> messages are composed of two sections: Attributes and a Prompt block (the request can contain several different Prompt blocks). The Attributes section includes several request-specific parameters. The Prompt block section itself is also composed of two sections: prompt-specific parameters and audio segments (audio / variable). The (optional) prompt-specific parameters include:

- *locale*: defines the language in which the prompt block is played (supported for local files only). For detailed information on language usage, refer to the *Audio Provisioning Server User's Manual* (LTRT-971xx).
- *baseurl*: defines a URL address that functions as a prefix to all audio segment URLs in the Prompt block.

The Prompt block contains references to one or more audio segments. The following audio segment types are available:

- **Physical Audio Segments:** These are physical audio files that are located either locally (on-blade) or on an external HTTP server. If the file is located on-blade, the reference to it is by using one of the following syntaxes:

  'file://x', 'file:///x', 'file:////x' or 'http://localhost/x'

  Where *x* stands for the file identifier (the ID or alias given by the APS server for local files; or the file's URL in for HTTP streaming).
- **Variables:** These are audio segments whose value is determined at run time. They are defined in the request as a <type, subtype, value> tuple. The device transforms the variable data to voice. To support variable playing, APS server support is mandatory. Available variable types are (subtypes in parenthesis): date (DMY - day month year; MDY - month day year - default), duration, month, money (USD), number (crd, ord), digit (gen, ndn) silence, string, time (t12, t24) and weekday.
  It is also possible to store audio files that are required to play supported types of phrases (e.g., dates and times) on an off-board system. This is beneficial in scenarios where the device's on-board storage limit has been reached, and thus, additional languages and audio can be stored off-board.
- **Sequences:** These are audio segments that consist of physical audio files and variables. These sequences can be defined using the APS server.

13.2.2.2 Operating with Audio Bundles

Voice prompts can be played from the device's local memory where they are stored as Audio bundles. An audio bundle is composed of a .dat file and an .xml file containing the information to properly parse the .dat file. Audio bundles are created through the APS and are then stored on a server supporting NFS or HTTP.

13.2.2.2.1 Uploading a Bundle to the Device

The audio bundle can be uploaded through FTP, NFS or HTTP. For more information, see the relevant Automatic Update chapter in the *Product Reference Manual*.

To upload a voice bundle to the device, the following *ini* file parameters should be set:

```
APSEnabled = 1
AMSProfile = 1
VpFileUrl = 'url-dat-file/dat-file'
APSSegmentsFileUrl = 'url-xml-file/xml-file''
```

Where *url-dat-file* and *url-xml-file* relate to the location of the relevant .dat and .xml files, and *dat-file* and *xml-file* relate to the file names, as shown in the example below:

```
APSEnabled = 1
AMSProfile = 1
VpFileUrl = 'http://10.50.2.1/dat files/vp.dat'
APSSegmentsFileUrl = 'http://10.4.2.5/segments/segments.xm'
```

For a detailed description of these *ini* file parameters, refer to the device's User's Manual.

You can upload a bundle to the device using one of the following methods:

- Loading an *ini* file as described above, and then resetting the device (hard reset). Optionally, you can configure parameters through Web interface or using SNMP, and then burn parameters to flash and reset the device through Web or SNMP (soft reset).
- Adding the following *ini* file parameter to periodically upload the .dat and .xml files:

```
AutoUpdateFrequency = 100          // updating is performed every
100 minutes.
```

  For more information, refer to the User's Manual regarding this *ini* file parameter and refer to Automatic Updates in the *Product Reference Manual*.
- Using SNMP to trigger an immediate upload of the files by setting acSysActionSetAutoUpdate to true.

Note: When uploading files through HTTP, if the names of the file that are already loaded to the device and the file intended to be uploaded are the same, time stamps of the old file and the new file should differ.

You can be notified on the outcome of an operation in two ways:

- Syslog messages – Informative Syslog messages are supplied when the operation has succeeded or failed. On operation failure, resort to first analyzing these messages.
- SNMP traps - Similar messages are also supplied via SNMP traps. For more information refer to the *Product Reference Manual*.

13.2.2.3 Playing Announcements

A <Play> request is used to play an announcement to the caller. Each <Play> request contains a single Prompt block and the following request-specific parameters:

- *id*: an optional random number used to synchronize request and response.
- *prompturl*: a specific audio file URL that is used in addition to the references in the Prompt block. This audio file is the first to be played.

An example of an MSCML <Play> Request that includes local and streaming audio files as well as variables is shown below:

```
<?xml version="1.0" encoding="utf-8"?>
<MediaServerControl version="1.0">
    <request>
          <play id=”123”>
                <prompt>
                      <audio url="http://localhost/1"/>
                      <variable type="digits" value="284"/>
                      <variable type="silence" value="1"/>
                      <audio url="http://10.3.0.2/aa.wav"/>
                <audiourl="nfs://10.3.0.3/prov data/bb.wav"/>
                </prompt>
          </play>
    </request>
</MediaServerControl>
```

13.2.2.4 Playing Announcements and Collecting Digits

The <PlayCollect> request is used to play an announcement to the caller and to then collect entered DTMF digits. The play part of the <PlayCollect> request is identical to the <Play> request. The collect part includes an expected digit map. The collected digits are continuously compared to the digit map. Once a match is found, the collected digits are sent in a <PlayCollect> response. The digit map should be in MGCP format (the type value must be set to ‘mgcpdigitmap’).

For example:

```
<regex type="mgcpdigitmap" value="([0-1]xxx)">
</regex>
```

Each <PlayCollect> request contains the following request-specific parameters in addition to the Prompt block (all parameters are optional):

- *id*: an optional random number used to synchronize request and response.
- *prompturl:* a specific audio file URL that is used in addition to the references in the prompt block. This audio file is the first to be played.
- *barge*: if set to ‘NO’, DTMF digits received during announcement playback are ignored. If set to ‘YES’, DTMF digits received during announcement playback stop the playback and start the digit collection phase.
- *firstdigittimer:* defines the amount of time (in milliseconds) the user does not enter any digits, after which a response is sent indicating timeout.
- *interdigittimer:* defines the amount of time (in milliseconds) the user does not enter any digits after the first DTMF digit is received, after which a response is sent indicating timeout.

- *extradigittimer:* used to enable the following:
  - Detection of command keys (ReturnKey and EscapeKey).
  - Not report the shortest match. MGCP Digitmap searches for the shortest possible match. This means that if a digitmap of (123 | 1234) is defined, once the user enters 123, a match is found and a response is sent. If ExtraDigitTimer is defined, the match can also be 1234 because the device waits for the next digits. To use ExtraDigitTimer, it must be defined in the request and you must add a "T" to the Digitmap (for example, 'xxT'). The ExtraDigitTimer is only used when a match is found. Before a match is found, the timer used is the InterDigitTimer. Therefore, if the ExtraDigitTimer expires, a "match" response reason is reported -- never a "timeout".
- *maxdigits:* defines the maximum number of collected DTMF digits after which the device sends a response.
- *cleardigits:* defines whether or not the device clears the digit buffer between subsequent requests.
- *returnkey*: defines a specific digit (including '*' and '#') which (when detected during a collection) stops the collection and initiates a response (that includes all digits collected up to that point) to be sent.
- *escapekey:* defines a specific digit (including '*' and '#') which (when detected during a collection) stops the collection and initiates a response (with no collected digits) to be sent.

An example is shown below of an MSCML <PlayCollect> Request that includes a sequence with variables and an MGCP digit map:

```
<?xml version="1.0" encoding="utf-8"?>
<MediaServerControl version="1.0">
          <request>
                <playcollect id="6379" barge="NO" returnkey="#">
                      <prompt>
                            <audio url="http://localhost/1">
                            <variable type="silence" value="1"/>
                            <variable type="date" subtype="mdy"
value="20041210"/>
                      </prompt>
                      <regex type="mgcpdigitmap" value="([0-
1]xxx)">
                      </regex>
                </playcollect>
          </request>
</MediaServerControl>
```

An example is shown below of an MSCML <PlayCollect> Response:

```
<?xml version="1.0" encoding="utf-8"?>
<MediaServerControl version="1.0">
          <response request="playcollect" id="6478" code="200"
text="OK" digits="4563">
          </response>
</MediaServerControl>
```

13.2.2.5 Playing Announcements and Recording Voice

The <PlayRecord> request is used to play an announcement to the caller and to then record the voice stream associated with that caller. The play part of the <PlayRecord> request is identical to the <Play> request. The record part includes a URL to which the voice stream is recorded. This URL refers to an HTTP server.

Each <PlayRecord> request contains the following request-specific parameters in addition to the Prompt block (all parameters except 'recurl' are optional):

- *id*: an optional random number used to synchronize request and response.
- *prompturl:* a specific audio file URL that is used in addition to the references in the prompt block. This audio file is the first to be played.
- *barge*: if set to 'NO', DTMF digits received during announcement playback are ignored. If set to 'YES', DTMF digits received during announcement playback stop the playback and start the recording phase.
- *cleardigits:* defines whether or not the device clears the digit buffer between subsequent requests.
- *escapekey:* defines a specific digit (including '*' and '#') which (when detected during any phase) stops the request and initiates a response.
- *recurl*: the URL on the external storage server to which the RTP stream is sent for recording. This is a mandatory parameter.
- *mode:* defines if the recording 'overwrites' the existing file or 'appends' to it.
- *initsilence:* defines how long to wait for initial speech input before terminating the recording. This parameter may take an integer value in milliseconds.
- *endsilence:* defines how long the device waits after speech has ended to stop the recording. This parameter may take an integer value in milliseconds.
- *duration*: the total time in milliseconds for the entire recording. Once this time expires, recording stops and a response is generated.
- *recstopmask:* defines a digit pattern to which the device compares digits detected during the recording phase. If a match is found, recording stops and a response is sent.

An example is shown below of an MSCML <PlayRecord> Request:

```
<?xml version="1.0" encoding="utf-8"?>
<MediaServerControl version="1.0">
          <request>
                <playrecord id="75899" barge="NO"
Recurl=nfs://10.11.12.13/save/recordings/11.wav>
                      <prompt>
                            <audio url="nfs://100.101.102.103/45">
                            <variable type="date" subtype="mdy"
value="20041210"/>
                      </prompt>
                </playrecord>
          </request>
</MediaServerControl>
```

An example is shown below of an MSCML <PlayRecord> Response:

```
<?xml version="1.0" encoding="utf-8"?>
<MediaServerControl version="1.0">
          <response request="playrecord" id="75899" code="200"
text="OK" reclength="15005">
          </response>
</MediaServerControl>
```

13.2.2.6 Stopping the Playing of an Announcement

The Application server issues a <stop> request when it requires that the device stops a request in progress and not initiate another operation. The only (optional) request-specific parameter is id.

The device refers to a SIP re-INVITE message with hold media (c=0.0.0.0) as an implicit <Stop> request. The device immediately terminates the request in progress and sends a response.

An example is shown below of an MSCML <Stop> Request:

```
<?xml version="1.0" encoding="utf-8"?>
<MediaServerControl version="1.0">
          <request>
                <stop id="123">
                </stop>
          </request>
</MediaServerControl>
```

13.2.2.7 Relevant Parameters

The following parameters (described in "IP Media Parameters" on page 363) are used to configure the MSCML:

- AmsProfile = 1 (mandatory)
- AASPackagesProfile = 3 (mandatory)
- VoiceStreamUploadMethod = 1 (mandatory)
- EnableVoiceStreaming = 1 (mandatory)
- MSCMLID (default=”ivr”)
- AmsPrimaryLanguage (default=”eng”)
- AmsSecondaryLanguage (default=”heb”)
- When using APS:
  - HeartBeatDestIP
  - HeartBeatDestPort
  - HeartBeatIntervalmsec
- When using AutoUpdate:
  - VPFileURL
  - APSSegmentsFileUrl
  - AutoUpdateFrequency / AutoUpdatePredefinedTime

13.2.2.8 Signal Events Notifications

The device supports Signal Events Notifications as defined in RFC 4722/5022 - MSCML. MSCML defines event notifications that are scoped to a specific SIP dialog or call leg. These events allow a client to be notified of various call progress signals. Subscriptions for call leg events are performed by sending an MSCML <configure_leg> request on the desired SIP dialog. Call leg events may be used with the MSCML conferencing or IVR services. Using the Signal Notifications, the device can report the following events:

Table 13-9: Reportable Events

Type	Subtype
AMD	▪ voice ▪ automata ▪ silence ▪ unknown
CPT	▪ SIT-NC ▪ SIT-IC ▪ SIT-VC ▪ SIT-RO ▪ busy ▪ reorder
FAX	▪ CED ▪ CNG ▪ modem

Below is an example:

```
<?xml version="1.0"?>
 <MediaServerControl version="1.0">
  <request>
   <configure_leg>
    <subscribe>
     <events>
      <signal type="amd" report="yes"/>
     </events>
    </subscribe>
   </configure_leg>
  </request>
 </MediaServerControl>


<?xml version="1.0"?>
 <MediaServerControl version="1.0">
  <notification>
   <signal type="amd" subtype="voice"/>
  </notification>
</MediaServerControl>
```

13.2.3 Voice Streaming

The voice streaming layer provides you with the ability to play and record different types of files while using an NFSor HTTP server.

13.2.3.1 Voice Streaming Features

The following subsections summarizes the Voice Streaming features supported on HTTP and NFS servers, unless stated otherwise.

13.2.3.1.1 Basic Streaming Play

You may play a *.wav, *.au or *.raw file from a remote server using G.711 coders.

13.2.3.1.2 Supported File Formats

The voice streaming layer provides support for *.wav, *.au, and *.raw file formats. The maximum supported header size of the file is 150 bytes.

In *.wav format, only mono mode and supported/known coders are supported. The maximum number of the non-data, non-fmt chunks can be up to 5.

13.2.3.1.3 Play from Offset

You may play a *.wav, *.au or *.raw file from a given offset within the file. Offset can be both positive and negative relative to the files length. A negative offset relates to an offset from the end of the file.

13.2.3.1.4 Remote File Systems

You may configure up to 16 remote file systems to operate with the system through NFS mounting.

13.2.3.1.5 Using Proprietary Scripts

You may use cgi or servlet scripts released with the version for recording to a remote HTTP server using the POST or PUT method.

13.2.3.1.6 Dynamic HTTP URLs

Voice streaming supports dynamic HTTP URLs. The following terminology is used:

- **Static audio content:** Traditional audio file URLs containing references to specific files (*.wav, *.au or *.raw). For example: http://10.50.0.2/qa/GOSSIP_ENG.wav
- **Dynamic audio content:** URLs referencing to cgi scripts or servlets. For example: http://10.50.0.2/cgi/getaudio.cgi?filename=DEFAULT_GREETING.raw&offset=0

In the case of dynamic URLs, the device performs the GET command with the supplied URL and as a result, the servlet or cgi script on the Web server is invoked. The Web server responds by sending a GET response containing the audio.

The URL format can be as follows (RFC 1738 URLs, section 3.3):

```
http://<host>:<port>/<path>?<searchpart>
```

where,

- **:<port>** is optional.
- **<path>** is a path to a server-side script.
- **<searchpart>** is of the form: key=value**[&key**=value]*

Note: At least one key=value pair is required.

Another example of a dynamic URL is shown below:

```
http://MyServer:8080/prompts/servlet?action=play&language=eng&file
=welcome.raw&format=1
```

(See also RFC 2396 URI: Generic Syntax.)

The servlet or cgi script can respond by sending a complete audio file or a portion of an audio file. The device skips any *.wav or *.au file header that it encounters at the beginning of the response. The device does not attempt to use any information in the header. For example, the device does not use the coder from the header. Note however, that the coder may be supplied through Web or *ini* file parameters.

13.2.3.1.7 Play LBR Audio File

You may play a file using low bit rate coders for *.wav and *.raw files.

Note: This feature is relevant for both NFS and HTTP.

13.2.3.1.8 Basic Record

You may record a *.wav, *.au or *.raw files to a remote server using G.711 coders.

Note: This feature is relevant for both NFS and HTTP.

13.2.3.1.9 Remove DTMF Digits at End of Recording

You may configure a recording to remove the DTMF received at the end, indicating an end of a recording.

Note: This feature is relevant for both NFS and HTTP.

13.2.3.1.10 Record Files Using LBR

You may record a file using low bit rate coders for *.wav and *.raw files.

Notes: This feature is relevant for both NFS and HTTP.

13.2.3.1.11 Modifying Streaming Levels Timers

Several parameters enable the user to control streaming level timers for NFS and HTTP and also the number of data retransmission when using NFS as the application layer protocol:

- **General command timeout – ServerRespondTimeout:** Defines the maximal time a command or respond may be delayed. This relates both to HTTP commands (GET, PUT, POST, HEAD etc.) and to NFS commands (create, lookup, read, write etc.).
- **Recording packet overruns timer – StreamingRecordingOverRunTimeout:** An overrun condition is one in which the device sends data to the server but does not receive responds from the server acknowledging that it received the data. Overruns relate to recording data to a remote server and result with “holes” in the recording. The streaming level aborts sessions containing consecutive overruns as derived from this timer. You may set the timer to longer periods than the default value, thereby enabling the device to be more "tolerant" to overrun conditions.
- **Playing packet underruns – StreamingPlayingUnderRunTimeout:** An underrun condition is one in which the device does not supply the DSPs with sufficient data, thus "starving" the DSPs. Underruns relate to playing data from a server to the device where, due to environmental conditions (usually network problems), the data is not passed quickly enough. This condition results with broken data passed to the user. The streaming level aborts sessions containing consecutive underruns as derived from this timer. You may set the timer to longer periods than the default value, thereby enabling the device to be more "tolerant" to underrun conditions.
- NFS command retransmission – NFSClientMaxRetransmission: Defines the number of times an NFS command is retransmitted when the server side does not respond. By default, the value is set to 0 and not used - instead, the number of retransmissions is derived from the server response timeout parameter and the current Recovery Time Objective (RTO) of the system.

These parameters may be configured using the *ini* file, Web interface, or SNMP.

13.2.3.2 Using File Coders with Different Channel Coders

The tables in the following subsections describe the support for different combinations of file coders (used for recording or playing a file) and channel coders (used when opening a voice channel).

The following abbreviations are used in the subsequent tables:

- **LBR:** Low Bit Rate Coder
- **PCMU:** G.711 µ-law coder
- **PCMA:** G.711 A-law coder
- **WB:** Linear PCM 16KHZ Wide Band Coder

Note: When recording with an LBR type coder, it is assumed that the same coder is used both as the file coder and the channel coder. Combinations of different LBR coders are currently not supported.

13.2.3.2.1 Playing a File

The table below lists the device's support of channel coders and file coders for playing a file.

Table 13-10: Coder Combinations - Playing a File

File Coder	File Type											
	WAV				AU				RAW			
	Channel Coder				Channel Coder				Channel Coder			
	PCMA	PCMU	LBR	WB	PCMA	PCMU	LBR	WB	PCMA	PCMU	LBR	WB
PCMA	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes
PCMU	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes
LBR	No	No	Yes	Yes	No	No	No	No	No	No	Yes	Yes

13.2.3.2.2 Recording a File

The table below lists the device's support of channel coders and file coders for recording a file.

Table 13-11: Coder Combinations - Recording a File

File Coder	File Type											
	WAV				AU				RAW			
	Channel Coder				Channel Coder				Channel Coder			
	PCMA	PCMU	LBR	WB	PCMA	PCMU	LBR	WB	PCMA	PCMU	LBR	WB
PCMA	Yes	Yes	Yes	No	Yes	Yes	Yes	No	Yes	Yes	Yes	No
PCMU	Yes	Yes	Yes	No	Yes	Yes	Yes	No	Yes	Yes	Yes	No
LBR	No	No	Yes	No	No	No	No	No	No	No	No	No

13.2.3.3 Maximum Concurrent Playing and Recording

For details on maximum concurrent playing and recording, refer to the relevant device's Release Notes.

13.2.3.4 LBR Coders Support

The following list describes the DSP templates required for using different low bit rate (LBR) coders and their support for *.wav, *.au, and *.raw files.

Table 13-12: DSP Templates

Coder	DSP Templates		
	*.wav file	*.raw file	*.au file
G.726 (Rate 16)	0-3	0-5	No
G.726 (Rate 24)	0-3	0-5	No
G.726 (Rate 32)	0-3	0-5	No
G.726 (Rate 40)	0-3	0-5	No
G.723.1 (Rate 5.3)	0	0	No
G.723.1 (Rate 6.3)	0	0	No
G.729	0-3	0-3	No
GSM FR	0,1	0-1	No
MS GSM	0,1	0-1	No
GSM EFR	0,1	1	No

Coder	DSP Templates		
	*.wav file	*.raw file	*.au file
AMR (Rate 4.75)	No	1	No
AMR (Rate 5.15)	No	1	No
AMR (Rate 5.9)	No	1	No
AMR (Rate 6.7)	No	1	No
AMR (Rate 7.4)	No	1	No
AMR (Rate 7.95)	No	1	No
AMR (Rate 10.2)	No	1	No
AMR (Rate 12.2)	No	1	No
QCELP (Rate 8)	No	2	No
QCELP (Rate 13)	No	2	No

13.2.3.5 HTTP Recording Configuration

The HTTP record method (PUT or POST) is configured using the following offline *ini* parameter:

```
// 0=post (default), 1=put
VoiceStreamUploadMethod = 1
```

The default value is shown below:

```
VoiceStreamUploadPostUri =
"/audioupload/servlet/AcAudioUploadServlet"
```

Note: The PUT method disregards this string.

13.2.3.6 NFS Configuration Using the *ini* File

An example of an NFS configuration is shown below. In this example, NFS server 192.168.20.26 shares two file systems - one rooted at /PROV_data, and the other rooted at /opt/uas. NFSv3 is used for both remote file systems. The defaults for UID(0) and GID(1) are used.

```
[NFSServers]
FORMAT NFSServers_Index = NFSServers_HostOrIP,
NFSServers RootPath, NFSServers NfsVersion;
NFSServers 0 = 192.168.20.26, /PROV data, 3;
NFSServers 1 = 192.168.20.26, /opt/uas, 3;
[\NFSServers]
```

Notes:

- The combination of Host/IP and Root Path should be unique for each row in the table. For example, there should be only one row in the table with a Host/IP of 192.168.1.1 and Root Path of /audio.
- To avoid terminating calls in progress, a row must not be deleted or modified while the system is accessing files on the remote NFS file system.
- An NFS file server can share multiple file systems. There must be a separate row in this table for each remote file system shared by the NFS file server that needs to be accessed by this system.
- For further details, refer to "Configuring the NFS Settings" on page 67.

13.2.3.7 Supported HTTP Servers

The following is a list of HTTP servers that are known to be compatible with AudioCodes voice streaming under Linux™:

- **Apache:** cgi scripts are used for recording and supporting dynamic URLs.
- **Jetty:** servlets scripts are used for recording and supporting dynamic URLs.
- **Apache tomcat:** using servlets.

13.2.3.7.1 Tuning the Apache Server

It is recommended to perform the following modifications in the http.conf file located in the apache conf/ directory:

- Define PUT script location: Assuming the put.cgi file is included in this package, add the following line for defining the PUT script (script must be placed in the cgi-bin/ directory):

```
Script PUT /cgi-bin/put.cgi
```

- Create the directory /the-apache-dir/perl (for example, /var/www/perl) and copy the CGI script to this directory. In the script, change the first line from c:/perl/bin/perl to your perl executable file (this step is required only if mod_perl is not included in your Apache installation).
- Keep-alive parameters: the following parameters must be set for correct operation with multiple POST requests:

```
KeepAlive On
MaxKeepAliveRequests 0 (unlimited amount)
```

- Using mode perl, fix the mod_perl to the following:

```
<IfModule mod perl.c>
<Location /cgi-bin>
 SetHandler perl-script
   PerlResponseHandler ModPerl::Registry
   Options +ExecCGI
   PerlOptions +ParseHeaders
   Order allow,deny
   Allow from all
</Location>
</IfModule>
```

- Apache MPM worker: it is recommended to use the Multi-Processing Module implementing a hybrid multi-threaded multi-process Web server. The following configuration is recommended:

```
<IfModule worker.c>
ThreadLimit         64
StartServers          2
ServerLimit         20000
MaxClients          16384
MinSpareThreads      100
MaxSpareThreads      250
ThreadsPerChild      64
MaxRequestsPerChild  16384
</IfModule>
```

13.2.3.8 Supporting NFS Servers

The table below lists the NFS servers that are known to be compatible with AudioCodes Voice Streaming functionality.

Table 13-13: Compatible NFS Servers

Operating System	Server	Versions
Solaris™ 5.8 and 5.9	nfsd	2, 3
Fedora™ Linux™ 2.6.5-1.358	nfsd	2, 3
Mandrake™ Linux™ v2.4.22	nfsd	2, 3
Windows™ 2000	Services For Unix™ (SFU)	2, 3
Windows™ 2000	winnfsd	2 (See Note)
SCO UnixWare™ 7.1.1	nfsd	2, 3
Windows™ 2000	Cygwin nfsd	2 (See Note)

Note: Cygwin and winnfsd support only NFSv2.

13.2.3.8.1 Solaris-Based NFS Servers

If you are using a Solaris™-based NFS server, then the following nfsd configuration modification is recommended, especially if you are planning to support voice recording:

- Edit the file /etc/default/nfs and set the value of NFSD_SERVERS to N*2, where *N* is the maximum number of record and play sessions that you expect to have in progress at any one time.

  The NFSD_SERVERS parameter controls the number of worker threads that the NFS daemon uses to satisfy requests. When a request arrives, a check is made for an idle worker thread. If an idle worker thread is available, then the request is passed to it. If an idle worker thread is unavailable, then a new one is created and the request is passed to it. If the limit in worker threads is reached, the request is queued until one of the existing worker threads is available. Queuing of NFS requests from a real-time application such as the media server should be avoided. Therefore, the NFSD_SERVERS parameter should be used to ensure there is an adequate number of worker threads.

  The default value for NFSD_SERVERS is 1. Typically, the /etc/default/nfs file contains NFSD_SERVERS set to 16.

  To determine how many worker threads are running on the NFS server, invoke the following command:

```
pstack `pgrep nfsd` | grep nfssys | wc -l
```

  An idle NFS daemon process displays 1 nfsd thread.

- Directories are shared by placing an entry in the /etc/dfs/dfstab file. See the share(1M) and share_nfs(1M) main pages for information on the format of entries in the dfstab file. Note that read-write (rw) is the default behavior. If you are planning to record to the file system, ensure that the directory is shared as rw. Also ensure that the recording directory has 777 (rwxrwxrwx) permissions.

  Below is an example /etc/dfs/dfstab file. Note that /audio1 is shared as read-only, and /audio2 is shared as read-write.

```
> cat /etc/dfs/dfstab
share -F nfs -o ro /audio1
share -F nfs /audio2
```

- Ensure that the /etc/nfssec.conf file is configured so that "sys" is the default security mode. You should see the following:

```
> cat /etc/nfssec.conf
none     0       -       -       -       # AUTH NONE
sys      1       -       -       -       # AUTH SYS
dh       3       -       -       -       # AUTH DH
default  1       -       -       -       # default is AUTH_SYS
```

- If the systems administrator wishes to use a default other than AUTH_SYS in the nfssec.conf file, then you should add "sec=sys" to each line in the dfstab file that is to be shared with an AudioCodes system. For example:

```
> cat /etc/dfs/dfstab
share -F nfs -o sec=sys,ro /audio1
share -F nfs -o sec=sys /audio2
```

- To restart the nfs daemon on Solaris, invoke the following two commands:

```
> /etc/init.d/nfs.server stop
> /etc/init.d/nfs.server start
```

- To view a log of directories which were shared on the previous restart of the nfs daemon, type the sharetab file. For example:

```
> cat /etc/dfs/sharetab
/audio1      -       nfs     ro
/audio2      -       nfs     rw
```

Other useful Solaris™ commands include the following:

- dfmounts: displays shared directories, including a list of clients that have these resources mounted.
- dfshares: displays a list of shared directories.

13.2.3.8.2 Linux-Based NFS Servers

The AudioCodes device uses local UDP ports that are outside of the range of 0..IPPORT_RESERVED(1024). Therefore, when configuring a remote file system to be accessed by an AudioCodes device, use the insecure option in the /etc/exports file. The insecure option allows the nfs daemon to accept mount requests from ports outside of this range.

Without the insecure option, the following nfs daemon log is received:

```
rpc.mountd: refused mount request from <ip> for <dir> illegal port
28000
```

Without the insecure option, the following Syslog is received:

```
NFS mount failed, reason=permission denied IP=<ip> path=<dir>
state=waitForMountReply numRetries=0
```

For more information, see the exports(5) main page on your Linux server.

An example /etc/exports entry is shown below:

```
/nfsshare *(rw,insecure,no_root_squash,no_all_squash,sync)
```

13.2.3.9 Common Troubleshooting

Always inspect the Syslog for any problem you may encounter; in many cases, the cause appears there.

Table 13-14: Troubleshooting

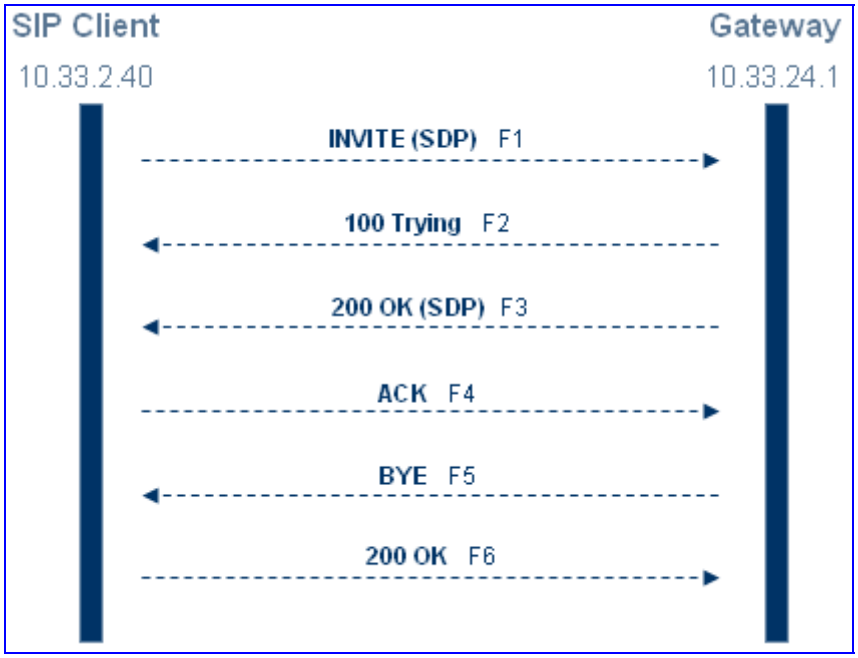
Problem	Probable Cause	Corrective Action
General Voice Streaming Problems		
Attempts to perform voice streaming operations results in each Syslog containing the string: 'VS_STACK_NOT_ACTIVE'.	Voice streaming is not enabled.	Enable voice streaming by loading an *ini* file containing this entry: EnableVoiceStreaming = 1
HTTP Voice Streaming Problems		
The last half-second of an announcement is not played, or a record operation terminates abnormally and the Syslog displays the following: 'VSReceiveFromNetwork: VS_CONNECTION_WITH_SERVER_LOST'. (The problem has been experienced with Apache version 2.0.50 on Solaris 9.)	The Web server is closing the virtual circuit at unexpected times.	Increase the Apache KeepAliveTimeout config parameter. Try to increase it to 30 seconds or longer than the longest announcement or expected record session.
NFS Voice Streaming Problems		
Announcement is terminated prematurelyand the Syslog displays the following: 'NFS request aborted … networkError'.	The AudioCodes media server has lost communication with the NFS server. A network problem or some problem with the NFS server exists.	Fix the network problem or NFS server problem. Ensure that the NFS server is not over-loaded.
Unable to play announcements from an NFS server and each Syslog displays the following: 'Unable to create new request, file system not mounted' 'NFS mount error …'	Either there is a problem with the NFS server, the network, or configuration of the media server or NFS server.	Fix the network problem or NFS server problem. Check the configuration on both the media server and the NFS server.
Record is terminated prematurely and the Syslog displays the following: 'VeData: no free buffers, req=16' 'Unable to play announNFS request aborted, reqid=16 cid=16 error=noRecordBufferError reqtype=vsHostRecord state=recTransfer'.	This occurs when the media server is receiving audio faster than it can save it to the remote NFS server. Either there is a problem with the NFS server, the network, or configuration of the media server or NFS server.	Fix the network problem or NFS server problem. Check the configuration on both the media server and the NFS server.

Problem	Probable Cause	Corrective Action
Remote file system is not being mounted and the Syslog displays the following: 'NFS mount failed, reason=permission denied IP=<ip> path=<dir> state=waitForMountReply numRetries=0;'.	The NFS server is not configured to accept requests on ports outside of the range 0...1024.	On a Linux NFS server, use the insecure option in the /etc/exports file (refer to Linux-Based NFS Servers on page 676).
All recording sessions are aborted at the same time with these Syslogs: 'NFS request aborted, reqid=209 cid=-1 error=writeReplyError reqtype=writeFile state=writeWait **[File:**NfsStateMachine.cpp …]' 'NFS request aborted, reqid=186 cid=-1 error=writeReplyError reqtype=writeFile state=writeWait **[File:**NfsStateMachine.cpp …]'	The file system on the NFS server is full.	Remove unwanted files on the file system.

13.2.4 Announcement Call Flow Example

The call flow, shown in the following figure, describes SIP messages exchanged between the device (10.33.24.1) and a SIP client (10.33.2.40) requesting to play local announcement #1 (10.8.25.17) using AudioCodes proprietary method.

Figure 13-1: Announcement Call Flow



1. **SIP MESSAGE 1:** 10.33.2.40:5060 -> 10.33.24.1:5060

```
INVITE sip:annc@10.33.24.1;play=http://10.3.0.2/hello.wav;repeat=2
SIP/2.0
Via: SIP/2.0/UDP 10.33.2.40;branch=z9hG4bKactXhKPQT
Max-Forwards: 70
From: <sip:103@10.33.2.40>;tag=1c2917829348
To: <sip:annc@10.33.24.1>
Call-ID: 1414622340oZZq@10.33.2.40
CSeq: 1 INVITE
Contact: <sip:103@10.33.2.40>
Supported: em,100rel,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
User-Agent: Audiocodes-Sip-Gateway-4.0 GA/v.4.0 GA
Content-Type: application/sdp
Content-Length: 215

v=0
o=AudiocodesGW 377662 728960 IN IP4 10.33.41.52
s=Phone-Call
c=IN IP4 10.33.41.52
t=0 0
m=audio 4030 RTP/AVP 4 0 8
a=rtpmap:4 g723/8000
a=rtpmap:0 pcmu/8000
a=rtpmap:8 pcma/8000
a=ptime:30
a=sendrecv
```

2. **SIP MESSAGE 2:** 10.33.24.1:5060 -> 10.33.2.40:5060

```
SIP/2.0 100 Trying
Via: SIP/2.0/UDP 10.33.2.40;branch=z9hG4bKactXhKPQT
From: <sip:103@10.33.2.40>;tag=1c2917829348
To: <sip:annc@10.33.24.1>;tag=1c1528117157
Call-ID: 1414622340oZZq@10.33.2.40
CSeq: 1 INVITE
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
Server: Audiocodes-Sip-Gateway-Mediant 1000/v.5.40.010.006D
Content-Length: 0
```

3. **SIP MESSAGE 3:** 10.33.24.1:5060 -> 10.33.2.40:5060

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.33.2.40;branch=z9hG4bKactXhKPQT
From: <sip:103@10.33.2.40>;tag=1c2917829348
To: <sip:annc@10.33.24.1>;tag=1c1528117157
Call-ID: 1414622340oZZq@10.33.2.40
CSeq: 1 INVITE Contact: <sip:10.33.24.1>
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
Server: Audiocodes-Sip-Gateway-Mediant 1000/v.5.40.010.006D
Content-Type: application/sdp
Content-Length: 165

v=0
o=AudiocodesGW 355320 153319 IN IP4 10.33.24.1
s=Phone-Call
c=IN IP4 10.33.24.1
t=0 0
```

```
m=audio 7170 RTP/AVP 0
a=rtpmap:0 pcmu/8000
a=ptime:20
a=sendrecv
```

4. **SIP MESSAGE 4:** 10.33.2.40:5060 -> 10.33.24.1:5060

```
ACK sip:10.33.24.1 SIP/2.0
Via: SIP/2.0/UDP 10.33.2.40;branch=z9hG4bKacnNUEeKP
Max-Forwards: 70
From: <sip:103@10.33.2.40>;tag=1c2917829348
To: <sip:annc@10.33.24.1>;tag=1c1528117157
Call-ID: 1414622340oZZq@10.33.2.40
CSeq: 1 ACK
Contact: <sip:103@10.33.2.40>
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
User-Agent: Audiocodes-Sip-Gateway-4.0 GA/v.4.0 GA
Content-Length: 0
```

5. **SIP MESSAGE 5:** 10.33.24.1:5060 -> 10.33.2.40:5060

```
BYE sip:103@10.33.2.40 SIP/2.0
Via: SIP/2.0/UDP 10.33.24.1;branch=z9hG4bKacFhtFbFR
Max-Forwards: 70
From: <sip:annc@10.33.24.1>;tag=1c1528117157
To: <sip:103@10.33.2.40>;tag=1c2917829348
Call-ID: 1414622340oZZq@10.33.2.40
CSeq: 1 BYE
Contact: <sip:10.33.24.1>
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
User-Agent: Audiocodes-Sip-Gateway-Mediant 1000/v.5.40.010.006D
Content-Length: 0
```

6. **SIP MESSAGE 6:** 10.33.2.40:5060 -> 10.33.24.1:5060

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.33.24.1;branch=z9hG4bKacFhtFbFR
From: <sip:annc@10.33.24.1>;tag=1c1528117157
To: <sip:103@10.33.2.40>;tag=1c2917829348
Call-ID: 1414622340oZZq@10.33.2.40
CSeq: 1 BYE
Contact: <sip:103@10.33.2.40>
Supported: em,timer,replaces,path
Allow:REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,IN
FO,SUBSCRIBE,UPDATE
Server: Audiocodes-Sip-Gateway-4.0 GA/v.4.0 GA
Content-Length: 0
```

13.3 Voice XML Interpreter

The device supports Voice Extensible Markup Language (VoiceXML) version 2.0. VXML is an XML-based scripting language used to prompt and collect information from callers. A VXML-based script may be used to control many types of interactive voice response (IVR) activities, including playing recorded announcements, collecting DTMF digits, recording a caller's voice, recognizing speech (i.e., automatic speech recognition or ASR), and synthesizing speech (i.e., text-to-speech or TTS). Its major goal is to bring the advantages of Web-based development and content delivery to interactive voice response applications.

Notes:

- VoiceXML is applicable only to Mediant 1000.
- Currently, automatic speech recognition (ASR) and text-to-speech (TTS) are not supported.

13.3.1 Features

VoiceXML offers the following features:

- VXML uses the AMS for enhanced audio features (i.e., playing prompts on a remote server, synthesized variables, enhanced digit patterns capabilities, different languages).
- Supports DTMF recognition.
- Supports recording of audio for later playback.
- Speech recognition: subscriber's speech is compared with voice grammars residing on an external speech server that is directed using the MRCP protocol) with matching words or phrases are returned as text strings.
- Text-to-Speech (TTS): regular text written in the IVR script is translated to speech and played to the user (the translation itself is done by an external server that is directed using the MRCP protocol).

13.3.2 Feature Key

VXML is protected by a Software Upgrade Key. However, if the device's Software Upgrade Key doesn't contain VXML, then VXML support is disabled. In such circumstances, upon trying to activate a VXML script, a Syslog message appears notifying the user that the VXML task was disabled. In addition, when the Software Upgrade Key doesn't contain VXML, the EnableVXML *ini* file parameteris ignored (i.e., although set to 1, VXML remains disabled).

13.3.3 VXML Scripts

Conceptually, there are two different types of VXML scripts that can be used (simultaneously or only one) by the device:

- **Dynamic scripts:** This script is downloaded as needed for an individual call and usually contains customized content for that particular call. When a call arrives, the device accesses a remote Web server to download a script. Once the script is downloaded. it's parsed, executed, and cleaned up at the end of the call.
- **Static scripts:** This script represents an application that can be used across many different callers. An example of such an application might be a drug prescription refill service where a prompt is played to the caller, the prescription number is obtained from the caller as speech or DTMF digits, and this data is then saved to an off-board database.

There are ramifications in using both these types of scripts. A dynamic script can be customized for each caller, but has to be downloaded and parsed for every call. However, static scripts are loaded once (although the system checks periodically for updates), parsed once, and then is re-used for each call. This results in better performance and using of fewer resources per call, because each call uses only what it needs of the parent script, and doesn't need its own copy of the full script.

Scripts are loaded initially through an INVITE message from the SIP Call Agent to the device. If the script is a static script, the device checks whether the file has already been loaded, and if so, it uses the existing script. Otherwise, the script is loaded to the device. Dynamic scripts are always loaded when requested.

A VXML script can trigger another VXML script to be loaded. An example of this is the VXML <goto> element, which can cause a transition to a different form in the same script, or to a completely different script. The script that is loaded as a result of the execution of the first script can be either dynamic or static. If the second script is a static script, the device checks whether it has already been loaded to the device and references that copy if it exists. Otherwise, the second script is loaded, parsed, and executed.

There are multiple ways in which VXML scripts may be loaded to the device. These include automatic update for static scripts (which allows for a script to be loaded using a remote FTP, HTTP, or NFS server), TFTP for static scripts (which allows a script to be loaded from a remote BootP/TFTP server), or HTTP for dynamic scripts. Refer to the appropriate sections for additional details.

13.3.4 Proprietary Extensions

To provide the functionality intended by the VXML specification and to extend the functionality of the VXML specification, some proprietary extensions have been included in the AudioCodes VXML Interpreter. These extensions are discussed in the following sections and are intended to enable a VXML script to make use of the advanced audio capabilities provided by the device.

13.3.4.1 Record

As the device doesn't provide the ability to record 'on-board', it is necessary to record a caller's speech by streaming the audio to either an external NFS server. There are two additional attributes for the VXML <record> element that can be used to specify the off-board file name as well as the streaming mechanism for recording speech.

- "dest" attribute for <record>, which refers to a fully specified URL. An example of this is the following script:

```
<?xml version="1.0"?>
<vxml version="2.0" xmlns=\"http://www.w3.org/2001/vxml\">
<form id="form1">
   <record name="msg" finalsilence="3000ms" maxtime="60s"
dtmfterm="true"
dest="http://192.168.1.2/recordings/greetings/callersspeech.wav">
      <audio src= "http://192.168.1.2/prompts/recordprompt.wav"/>
      <filled>
           <audio src = "http://192.168.1.2/prompts/confirm.wav"/>
           <audio src= "http://192.168.1.2/
greetings/callersspeech.wav"/>
         <exit/>
      </filled>
       <noinput>
           <audio src=
"http://192.168.1.2/prompts/recordprompt2.wav "/>
            <reprompt/>
        </noinput>
   </record>
  </form>
</vxml>
```

In this example, the "dest" attribute of the <record> element specifies that the caller's speech must be streamed with HTTP to the system with IP address 192.168.1.2, and stored in a file called "callersspeech.wav" on that system.

- The "destexpr" attribute provides an alternative to the "dest" attribute. The "destexpr" attribute is evaluated during runtime to determine where to store the caller's speech. The following is an example script illustrating its usage:

```
<?xml version="1.0"?>
<vxml version="2.0" xmlns=\"http://www.w3.org/2001/vxml\">
<var name="recordpath" expr =
"'http://192.168.1.2/recordings/greetings/'"/>
<form id="form1">
   <record name="msg" finalsilence="3000ms" maxtime="60s"
dtmfterm="true" destexpr="recordpath + 'callersspeech.wav'">
      <audio src= "http://192.168.1.2/prompts/recordprompt.wav"/>
      <filled>
           <audio src = "http://192.168.1.2/prompts/confirm.wav"/>
           <audio expr= " recordpath + 'callersspeech.wav'"/>
         <exit/>
      </filled>
       <noinput>
           <audio src=
"http://192.168.1.2/prompts/recordprompt2.wav"/>
            <reprompt/>
        </noinput>
   </record>
  </form>
</vxml>
```

13.3.4.2 Audio Extensions

The device provides a rich set of functionality for building and playing announcements using recorded audio files. This functionality includes the ability to play certain types of phrases such as date, time, and number based upon a specific languages grammar rules. The files used to build the announcements can be stored on the device, or can be stored off-board on an external file system.

To take advantage of the advanced announcement capabilities provided by the device, the AudioCodes resident VXML Interpreter provides some extensions to the VXML <audio> element. These extensions are discussed in the following sections.

For more information on provisioning audio for the device, please refer to the *Audio Provisioning Server (APS) User's Manual*.

13.3.4.2.1 Local Audio

While not a true extension, it's possible to play audio files that reside on-board a device. The following is an example of how such an audio file can be referenced using a VXML <audio> element.

```
<audio src = "http://localhost/123"/>
```

This reference directs the VXML software to play the audio segment marked with identifier '123'.

Using this method of access, the advanced audio structures defined by the AudioCodes Audio Provisioning Server (APS) can be referenced. While these various structures are outside the scope of the current document, they include sets, sequences, and multi-language variables. For more information on these advanced audio structures, please refer to the *Audio Provisioning Server (APS) User's Manual*.

13.3.4.2.2 Say-as Tag for the Audio Element

While the VXML <say-as> tag is typically used as a directive to a text-to-speech engine in association with a VXML <prompt> element, the AudioCodes resident VXML Interpreter allows the <say-as> tag to also be used with the <audio> element. In this context, the <say-as> tag directs the VXML Interpreter to play phrases such as dates and times using provisioned audio files. The following is an example of an <audio> element using the <say-as> extension:

```
<audio> <say-as interpret-as="date"> 20080704 </say-as> </audio>
```

This example assumes that the device has been provisioned with the appropriate audio to play this example. The <audio> element in the example directs the VXML Interpreter to announce the date "July 4th, 2008".

The following table lists the supported phrase types, any valid subtypes for the phrases, the expected input format for each phrase type, and any notes for the various phrase types.

Table 13-15: Say-as Phrase Types

Say-as Token	Variable Type	Variable Subtype	Variable Input Format	Note
date	date	None supported	yyyymmdd	Dates are always announced according to the grammar rules of the

Say-as Token	Variable Type	Variable Subtype	Variable Input Format	Note
				language.
duration	duration	None supported	The input is up to 10 digits, with the value representing the duration in seconds.	Duration is always announced as hours, minutes, and seconds.
currency	money	Three-character ISO currency code. There is a specific set of currencies supported by the device, which are documented in Audio Provisioning Server (APS) User's Manual: Audio Files.	The input is up to 10 digits.	The number is converted appropriately to the currency in question. A value of 1234 in US Dollars, for example, is spoken as 12 dollars and 34 cents. The same input as Yen would be 1 thousand, 234 Yen.
number	number	cardinal	Up to 10 digits.	Integer
number:cardinal	number	cardinal	Up to 10 digits.	
number:ordinal	number	ordinal	Up to 10 digits.	Range of supported ordinal numbers varies by language, as to whether a certain ordinal number is supported by the language.
number:digits	digits	generic	A string of up to 64 digits including 0-9, * and #.	
telephone	digits	generic	A string of up to 64 digits including 0-9, * and #.	
telephone:ndn	digits	North American DN	Must be 10 digits 0-9.	
telephone:gen	digits	generic directory number	A string of up to 64 digits including 0-9, *, and #.	
time	time	t24	hhmm	24 hour time.
time:t12	time	t12	hhmm	
time:t24	time	t24	hhmm	

Below are two examples that direct the device to announce the cardinal number 1000.

```
<audio> <say-as interpret-as="number"> 1000 </say-as> </audio>
<audio> <say-as interpret-as="number:cardinal"> 1000 </say-as>
</audio>
```

In the example below, the device is directed to announce the string as a North American directory number. The output is “eight hundred, five five five, one two one two”. A few moments of silence are inserted at the points in the phrase indicated by commas.

```
<audio> <say-as interpret-as=”telephone:ndn”> 8005551212 </say-as>
</audio>
```

In the example below, the device outputs the announcement “one million two hundred twenty thousand seven hundred dollars and fifteen cents”:

```
<audio> <say-as interpret-as=”currency:usd”> 122070015 </say-as>
</audio>
```

13.3.4.2.3 Supplying Values to Provisioned Variables

As mentioned previously, the APS provides the capability to provision several types of advanced audio structures, including multi-language variables. A multi-language variable is an instance of one of the supported phrase types such as date and time. The APS assigns a numeric segment identifier to each variable, and the value for the variable can be provided at runtime. VXML doesn't define any capability for passing a value to a variable, therefore, the AudioCodes VXML Interpreter provides an extension to support this capability.

Below is an example that demonstrates this capability. Assume that a variable of type “date” has been provisioned on the APS, and the variable has been assigned segment identifier 17.

```
<audio src=”http://localhost/17?var=20080120”/>
```

In this example, the device outputs the date “January 20th, 2008”.

13.3.4.2.4 Supplying Selector Values to Provisioned Variables and to Say-as Phrases

Another concept supported by the device is “selectors”. A selector is a keyword and value pair that is used by the device software to build announcements. There can be many combinations of keywords and values used for selectors, but the keyword “lang” and a language code are especially useful because this pair of tokens can be used to vary the language for announcements. For example, to announce the date from the previous example in French, below is syntax using the previous example along with a selector that builds the French announcement:

```
<audio src=”http://localhost/17?var=20070620&sel=lang=fr”/>
```

More than one selector in an <audio> element can exist. In the example below, the language is French and gender selector with the value “female” is also specified:

```
<audio
src=”http://localhost/17?var=20070620&sel=lang=fr&gender=F”/>
```

Selectors can also be useful in combination with <say-as> elements. For example, the following illustrates making the same announcement from the previous example using a <say-as> element:

```
<audio src="?sel=lang=fr&gender=F"> <say-as interpret-as="date">
20070620 </say-as> </audio>
```

For more information regarding available, selectors, please refer to the *Audio Provisioning Server (APS) User's Guide*.

13.3.4.3 Language Identifier Support

The AudioCodes resident VXML engine supports language identifiers as specified by RFC 3066. However, when accessing audio resident on the device using the proprietary extensions described earlier, the country code portion of the identifier is ignored. In addition, the language code portion of the identifier supports the languages listed in the table below.

Table 13-16: Support for Language Code Portion of Identifier

Language Code	Language
bd	Belgian Dutch
ca	Catalan
cs	Czech
de	German
el	Greek
en	English
es	Spanish
eu	Basque
fr	French
gl	Gallegan
he	Hebrew
hi	Hindi
it	Italian
ja	Japanese
ko	Korean
ms	Malay
nl	Netherlands Dutch
pt	Portuguese
ru	Russian
sw	Swedish
th	Thai
tl	Tagalog

Language Code	Language
tr	Turkish
vi	Vietnamese
yu	Cantonese
zh	Mandarin

13.3.5 Combining <audio> Elements

The VXML specification supports multiple <audio> elements nested within other elements such as prompts. An example demonstrating this functionality which includes the AudioCodes extensions is useful to show how multiple components can be combined to create a single announcement.

The following example shows how an announcement can be constructed that says "Welcome to Acme Corporation. Today's date is June 20th, 2008. Today's special is large widgets, two for ten dollars." For the sake of the example, assume the following:

- "Welcome to Acme Corporation" and "Today's special is large widgets, two for ten dollars" are stored on an external file system and is played using HTTP streaming.
- "Today's date is" is a recording provisioned on the APS as segment 99.
- "June 20th, 2008" is a multi-language variable announcement made up of multiple recordings provisioned on the APS.

```
<prompt bargein="false">
   <audio src = "http://192.168.1.2/announcements/welcome.wav"/>
   <audio src = "http://localhost/99"/>
   <audio src="?sel=lang=en"> <say-as interpret-as="date">
20080620 </say-as> </audio>
   <audio src =
"http://192.168.1.2/announcements/todaysspecial.wav"/>
</prompt>
```

13.3.6 Notes Regarding Non-compliant Functionality

The AudioCodes resident VXML Interpreter doesn't asynchronously throw events as described in the VXML Specification. For example, if the clear element tries to clear an element within a script and that element does not exist, the VXML Specification specifies that the Interpreter should throw an "error.badfetch" event. In contrast, the AudioCodes Interpreter logs an appropriate error to the syslog for the device and the script exits. The VXML Interpreter behaves similarly for software errors in general such as running out of memory resources, trying to access non-existent audio files, etc. The impact of not throwing events asynchronously can be minimized by carefully testing all code paths for a VXML script before its deployment.

13.3.7 Supported Elements and Attributes

The following status legend should be referenced for all tables in the following subsections:

- **NS:** Not Supported
- **PS:** Partially Supported
- **S:** Supported

13.3.7.1 VoiceXML Supported Elements and Attributes

Table 13-17: VoiceXML Supported Elements and Attributes

Element	Parameter	Max Size	Shadow Variable	Status	Comments
<assign>				S	
	name	64		S	
	expr	128		S	
<audio>				S	The AudioCodes audio element has proprietary extensions in addition to attributes from the standard to support on-board audio variables.
	src	256		S	
	fetchtimeout			NS	
	fetchhint			NS	Default behavior is "safe"; fetch document when it's needed.
	maxage			NS	
	maxstale			NS	
	expr	128		S	
	caching			Ignored	1.0 VXML attribute not present in VXML 2.0.
<block>				S	
	name	32		S	
	expr	128		S	
	cond	64		S	
<catch>				S	
	event	64		S	
	count	numeric		S	
	cond	128		S	
<choice>				S	
	dtmf			S	
	accept			NS	
	next	256		S	
	expr	128		S	
	event	64		S	

Element	Parameter	Max Size	Shadow Variable	Status	Comments
	eventexpr	128		S	
	message			NS	
	messageexpr			NS	
	fetchaudio			NS	
	fetchtimeout			NS	
	fetchhint			NS	Default behavior is "safe"; fetch document when it's needed.
	maxage			NS	
	maxstale			NS	
<clear>				S	
	namelist	4 * 32		S	
<disconnect>				S	
<else>				S	
<elseif>				S	
	cond	128		S	
<enumerate>				NS	
<error>				S	
	count	numeric field		S	
	cond	128		S	
<exit>				S	
	expr	128		S	
	namelist	4 * 32		S	
<field>				S	
	name	32		S	
	expr	128		S	
	cond	128		S	
	type	enum		PS	Built-in grammars are supported for recognition against fields, but the match isn't spoken as the built-in type in text-to-speech.
	slot			NS	Default value is the variable name, thus, slot is not needed.
	modal	true/false		S	

Element	Parameter	Max Size	Shadow Variable	Status	Comments
		64	name$.utterance	S	
		enum	name$.inputmode	S	
		64	name$.interpretation	S	
		numeric	name$.confidence	S	
<filled>				S	
	mode			S	
	namelist	4 * 32		S	
<form>				S	
	id	32		S	
	scope	enum		S	
<goto>				S	
	next	256		S	
	expr	128		S	
	nextitem	32		S	
	expritem	128		S	
	fetchaudio			NS	
	fetchtimeout			NS	
	fetchhint			NS	
	maxage			NS	
	maxstale			NS	
<grammar>				S	
	version			S*	For voice grammars, this is passed to the speech recognition engine.
	xml:lang	5		S*	For voice grammars, this is passed to the speech recognition engine.
	mode			S	
	root			S	
	tag			S	
	xml:base			NS	

Element	Parameter	Max Size	Shadow Variable	Status	Comments
	src	256		S	
	scope	enum		S*	In this release, a document scope grammar isn't active in a dialog scope form.
	type	enum		PS	Built-in grammars are supported for recognition against fields, but the match is not spoken as the built-in type in text-to-speech.
	weight	numeric		S*	For voice grammars, this is passed to the speech recognition engine.
	fetchtimeout			NS	Voice grammars are maintained on the speech recognition server, not on device, thus this set of attributes that control caching of grammar doesn't apply.
	fetchhint			NS	Voice grammars are maintained on the speech recognition server, not on device, thus this set of attributes that control caching of grammar doesn't apply.
	maxage			NS	Voice grammars are maintained on the speech recognition server, not on device, thus this set of attributes that control caching of grammar doesn't apply.
	maxstale			NS	Voice grammars are maintained on the speech recognition server, not on device, thus this set of attributes that control caching of grammar doesn't apply.
<help>				S	
	count	numeric field		S	
	cond	128		S	
<if>				S	
	cond	128		S	
<initial>				NS	The initial element and all its attributes aren't supported in this release.
	name			NS	
	expr			NS	
	cond			NS	
<link>				S	
	next	256		S	
	expr	128		S	
	event	65		S	
	eventexpr	128		S	

Element	Parameter	Max Size	Shadow Variable	Status	Comments
	message			NS	
	messageexpr			NS	
	dtmf	31		S	
	fetchaudio			NS	
	fetchtimeout			NS	
	fetchhint			NS	Default behavior is "safe"; fetch document when it's needed.
	maxage			NS	
	maxstale			NS	
<log>				S	
	label	32		S	
	expr	128		S	
<menu>				S	
	id	32		S	
	scope	enum		S	
	dtmf	true/false		S	
	accept			NS	It's not obvious how to instruct the speech recognition engine that approximate matches are acceptable.
<noinput>				S	
	count	numeric		S	
	cond	128		S	
<nomatch>				S	
	count	numeric field		S	
	cond	128		S	
<object>				S	
	name	*		S	Since objects are developed for proprietary purposes as needed, attribute sizes aren't listed.
	expr			S	
	cond			S	
	classid			S	

Element	Parameter	Max Size	Shadow Variable	Status	Comments
	codebase			S	
	codetype			S	
	data			S	
	type			S	
	archive			S	
	fetchtimeout			NS	
	fetchhint			NS	Default behavior is "safe"; fetch document when it's needed.
	maxage			NS	
	maxstale			NS	
<option>				S	
	dtmf	31		S	
	accept			NS	It's not obvious how to instruct the speech recognition engine that approximate matches are acceptable.
	value	32		S	
<param>				S	
	name	32		S	
	expr	128		S	
	value	128		S	
	valuetype	enum		S	
	type	128		S	
<prompt>				S	
	bargein	true/false		S	
	bargeintype			PS	Speech barge-in is supported, but not hotword.
	cond	128		S	
	count	numeric		S	
	timeout	numeric		S	
	xml:lang	5			
	xml:base	256			
<property>				S	
	name	32		S	

Element	Parameter	Max Size	Shadow Variable	Status	Comments
	value	128		S	
<record>				S	
	name	32		S	
	expr	128		S	
	cond	128		S	
	modal			NS	Grammars are not supported, thus, modal doesn't apply.
	beep	true/false		S	Requires that a user-defined tone be added to the system. Please refer to "Example of UDT 'beep' Tone Definition" on page 703 for an example. Please refer to the Auxilary Files section for additional details regarding creating user-defined tones.
	maxtime	time value		S	
	finalsilence	time value		S	
	dtmfterm			NS	DTMF and voice grammars aren't supported for record, but the termchar property can be used to terminate recordings.
	type			NS	The recorded audio format is specified by the file extension in the dest or destexpr attribute.
	dest	256		S*	Not part of the standard, either this attribute or destexpr are needed to specify the remote URL where the recorded audio is stored.
	destexpr	128		S*	Refer to previous item.
		numeric	name$.duration	S	
			name$.size	NS	As recorded audio is not stored onboard the device size is not available.
		1	name$.termchar	S	
		true/false	name$.maxtime	S	
<reprompt>				S	
<return>				S	
	event	64		S	
	eventexpr	128		S	
	message			NS	
	messageexpr			NS	

Element	Parameter	Max Size	Shadow Variable	Status	Comments
	namelist	4 * 32		S	
<script>				NS	The script element and all of its attributes are not supported.
	src			NS	
	charset			NS	
	fetchtimeout			NS	
	fetchhint			NS	
	maxage			NS	
	maxstale			NS	
<subdialog>				S*	Playing a prompt from a sub-dialog element is not supported in this release.
	name	32		S	
	expr	128		S	
	cond	128		S	
	namelist	4 * 32		S	
	src	256		S	
	srcexpr	128		S	
	method	enum		S	
	enctype			NS	
	fetchaudio			NS	
	fetchtimeout			NS	
	fetchhint			NS	Default behavior is "safe"; fetch document when it's needed.
	maxage			NS	
	maxstale			NS	
<submit>				S	
	next	256		S	
	expr	128		S	
	namelist	4 * 32		S	
	method	enum		S	
	enctype			NS	
	fetchaudio			NS	

Element	Parameter	Max Size	Shadow Variable	Status	Comments
	fetchtimeout			NS	
	Fetchhint			NS	Default behavior is "safe"; fetch document when it's needed.
	Maxage			NS	
	maxstale			NS	
<throw>				S	
	Event	64		S	
	eventexpr	128		S	
	message			NS	
	messageexpr			NS	
<transfer>				S	
	Name			S	
	Expr			NS	
	Cond			S	
	Dest			NS	Only numbers.
	destexpr			S	
	Bridge			S	Only blind transfer supported (false).
	type			S	Only blind transfer supported (blind).
	connecttimeout			NS	
	maxtime			NS	
	transferaudio			NS	
	Aai			NS	
	Aaiexpr			NS	
			name$.duration	S	
			name$.inputmode	S	
			name$.utterance	S	
<value>				S	
	expr	128		S	
<var>				S	
	name	32		S	

Element	Parameter	Max Size	Shadow Variable	Status	Comments
	expr	128		S	
<transfer>	Name			S	
	Expr			NS	
	Cond			S	
	Dest			NS	Only numbers.
	destexpr			S	
	Bridge			NS	Only Bridge = false
	type			NS	Only type = blind
	connecttimeout			NS	
	maxtime			NS	
	transferaudio			NS	
	Aai			NS	
	Aaiexpr			NS	
			name$.duration	S	
			name$.inputmode	S	
			name$.utterance	S	
<value>				S	
	expr	128		S	
<var>				S	
	name	32		S	
	expr	128		S	
<vxml>				S	

13.3.7.2 SRGS and SSML Support

Note that elements associated with either the Speech Recognition Grammar Specification (SRGS) or Speech Synthesis Markup Language (SSML) are used to control the behavior of a remote speech engine for either speech recognition or text-to-speech. These elements would be passed from the VXML interpreter to the remote speech engine and are outside the scope of VXML.

13.3.7.3 VoiceXML Supported Properties

Table 13-18: VoiceXML Supported Properties

Platform Properties	Status	Equivalent *ini* file parameter or Notes
Recognizer		
confidencelevel	S	VxmlConfidenceLevel
Sensitivity	S	VxmlSensitivityLevel
speedvsaccuracy	S	VxmlSpeedVsAccuracy
Completetimeout	S	VxmlCompleteTimeout
incompletetimeout	S	VxmlInCompleteTimeout
maxspeechtimeout	S	VxmlMaxSpeechTimeout
DTMF Recognizer		
Interdigittimeout	S	VxmlInterDigitTimeout
Termtimeout	S	VxmlTermTimeout. Note that the system default is not 0 as directed in the specification for the protocol, but 3 seconds. This is to ensure digit collection functions correctly.
Termchar	S	VxmlTermChar
Prompt and Collect		
Bargein	S	VxmlBargeinAllowed
Bargeintype	NS	Regular speech vs hotword bargein
Timeout	S	VxmlNoInputTimeout
Fetching		
Audiofetchhint	NS	
Audiomaxage	NS	
Audiomaxstale	NS	
documentfetchhint	NS	
documentmaxage	NS	
documentmaxstale	NS	
grammarfetchhint	NS	
Grammarmaxage	NS	
Objectfetchhint	NS	
Objectmaxage	NS	
Objectmaxstale	NS	

Platform Properties	Status	Equivalent *ini* file parameter or Notes
Scriptfetchhint	NS	
Scriptmaxage	NS	
Scriptmaxstale	NS	
Fetchaudio	NS	
Fetchaudiodelay	NS	
fetchaudiominimum	NS	
Fetchtimeout	NS	
Miscellaneous		
Inputmodes	S	VxmlSystemInputModes. Note that the system default is 0 (DTMF) vs 2 (Voice and DTMF) as specified in the specification. This is because the majority of systems are expected to use DTMF collection and local or streamed announcements as opposed to text-to-speech and speech recognition.
Universals	NS	Universal grammars and behaviors such as help, cancel, and exit. Default is none.
Maxnbest	NS	Size of last result array

13.3.7.4 VoiceXML Variables and Events

Table 13-19: VoiceXML Variables and Events

Variable/Event Name	Status	Notes
Standard Session Variables		
session.connection.local.uri	NS	
session.connection.remote.uri	NS	
session.connection.protocol.name	NS	
session.connection.protocol.version	NS	
session.connection.redirect	NS	
session.connection.aai	NS	
session.connection.originator	NS	
Standard Application Variables		
application.lastresult$	S	The application.lastresult variables array is one element deep.
application.lastresult$**[i]**.confidence	S	
application.lastresult$**[i]**.utterance	S	
application.lastresult$**[i]**.inputmode	S	
application.lastresult$**[i]**.interpretation	S	
Pre-defined Events		

Variable/Event Name	Status	Notes
Note: while throwing and catching events from scripts are supported, throwing events asynchronously from within the interpreter (e.g., an event.badfetch) is currently not supported.		
catch	S	
connection.disconnect.hangup	NS	
connection.disconnect.transfer	NS	
exit	S	
help	S	
noinput	S	
nomatch	S	
maxspeechtimeout	S	
error.badfetch	PS	In most cases, the conditions that would cause this event are recognized during script parsing, thus, the script loading fails.
error.badfetch.http.response_code	NS	
error.badfetch.protocol.response_code	NS	
error.semantic	PS	
error.noauthorization	NS	
error.noresource	NS	
error.unsupported.builtin	NS	
error.unsupported.format	NS	
error.unsupported.language	NS	
error.unsupported.objectname	NS	Unsupported elements are recognized during initial parsing, thus, the script isn't executed, and no events are thrown.
error.unsupported.element	NS	Unsupported elements are recognized during initial parsing, thus, the script isn't executed, and no events are thrown.
Transfer Events		
connection.disconnect.hangup	NS	
connection.disconnect.transfer	NS	
Transfer Errors		
error.connection.noauthorization	NS	
error.connection.baddestination	NS	
error.connection.noroute	NS	
error.connection.noresource	NS	
error.connection.protocol.nnn	NS	
error.unsupported.transfer.blind	NS	

Variable/Event Name	Status	Notes
error.unsupported.transfer.bridge	NS	
error.unsupported.uri	NS	

13.3.7.5 ECMAScript Support

The following table describes the ECMAScript support that the AudioCodes resident VXML engine provides. As shown in the example below, all operands and operators in an expression must be separated by one or more ECMAScript whitespace characters.

```
<var name="orange" expr="var1 + 7"/>
```

Below is an example of incorrect formatting (i.e., not supported):

```
<var name="orange" expr="var1+7"/>
```

Table 13-20: ECMAScript Support

Operand/Operator	Examples	Status	Note
Whitespace chars	tab, vertical tab, form feed, and space	S	
Arithmetic Operators	+, ++, -, --, *, /, %	S	
Logical Operators	&&, ||, !	S	
Assignment Operators	=, +=, -=, *=, /=, %=, &=, ^=, |=, <<=, >>=, >>>=	S	
Bitwise Operators	&, ^, |, ~, <<, >>, >>>	S	
Comparison Operators	==, !=, >, >=. <. <=	S	
String Operators	+, +=	S	
Entity Reference Mapping The following is supported / required: **Operator Entity Reference** < &lt <= <= > >= >= >= && && Support for the '≤' and '≥' entities is currently not available.		S	
Null Literals	null	S	Section 7.8.1, ECMA-262 3rd Edition December, 1999
Boolean Literals	true, false	S	Section 7.8.2, ECMA-262 3rd

Operand/Operator	Examples	Status	Note
			Edition December, 1999
Numeric Literals		S	Section 7.8.3, ECMA-262 3rd Edition December, 1999
String Literals		S	Section 7.8.4, ECMA-262 3rd Edition December, 1999

13.3.8 Example of UDT 'beep' Tone Definition

The following is an example definition for 'beep' tone used for the <record> element:

```
#record beep tone
[CALL PROGRESS TONE #1]
Tone Type=202
Low Freq [Hz]=430
High Freq [Hz]=0
Low Freq Level [-dBm]=13
High Freq Level [-dBm]=0
First Signal On Time [10msec]=100
First Signal Off Time [10msec]=0
Second Signal On Time [10msec]=0
Second Signal Off Time [10msec]=0
Default Duration [msec]=350
```

13.3.9 Limitations and Restrictions

The maximal length of the VXML file is 65536 bytes.

Reader's Notes

14 SIP Software Package

The table below lists the device's standard SIP software package.

Table 14-1: Software Package

File Name	Description
Firmware (RAM CMP) File	
M600_Digital_SIP_xxx.cmp	Image file containing the software for the Mediant 600 device.
M1000_Digital_SIP_xxx.cmp	Image file containing the software for the Mediant 1000 device.
M1000_SIP_xxx.cmp	Image file containing the software for both FXS and FXO modules.
***ini* Configuration Files**	
M600_Digital_SIP_T1.ini	Sample *ini* file for Mediant 600 E1 devices.
M600_Digital_SIP_E1.ini	Sample *ini* file for Mediant 600 T1 devices.
SIPgw_M1K.ini	Sample *ini* file for the Mediant 1000 and Mediant 600 devices.
M1000_Digital_SIP_T1.ini	Sample *ini* file for Mediant 1000 E1 devices.
M1000_Digital_SIP_E1.ini	Sample *ini* file for Mediant 1000 T1 devices.
Usa_tones_xx.dat	Default loadable Call Progress Tones *.dat file
Usa_tones_xx.ini	Call Progress Tones *ini* file (used to create *.dat file)
Utilities	
DConvert	TrunkPack Downloadable Conversion Utility - to create Call Progress Tones, Voice Prompts, and CAS files
ACSyslog	Syslog server
BootP	BootP/TFTP configuration utility
CPTWizard	Call Progress Tones Wizard
CAS Protocol Files	Used for various signaling types, such as E_M_WinkTable.dat
MIB Files	MIB library for SNMP browser
CAS Capture Tool	Utility that is used to convert CAS traces to textual form
ISDN Capture Tool	Utility that is used to convert ISDN traces to textual form

Notes:

- The *ini* and Utility files are shipped with the device in CD format.
- The device is supplied with a cmp file pre-installed on its flash memory. Therefore, this file is not included on the supplied CD. However, if you are an AudioCodes registered customer, you can obtain the latest cmp version files (as well as documentation and other software such as the *ini* and MIB files, and Utilities) from AudioCodes Web site at www.audiocodes.com/support (customer registration is performed online at this Web site). If you are not a direct customer of AudioCodes, please contact the AudioCodes' Distributor and Reseller from whom this product was purchased.

Reader's Notes

15 Selected Technical Specifications

15.1 Mediant 1000

The table below lists the main technical specifications of the Mediant 1000.

Table 15-1: Mediant 1000 Functional Specifications

Function	Specification
Interfaces	
Modularity and Capacity	Voice interface: Equipped with 6 Slots that can host voice modules. Up to a maximum of 24 analog ports or 4 digital spans.
Digital Modules	1, 2 or 4 E1/T1/J1 spans using RJ-48c connectors per module. Up to 4 digital modules (maximum 4 spans per gateway). Optional 1+1 or 2+2 fallback spans.
Analog FXO and FXS Modules	4 ports using RJ-11 connectors per module; Up to 6 modules per gateway, Ground Start and Loop Start.
BRI Module	4 BRI ports (8 calls) per module, up to 5 modules per gateway with S/T interfaces. Supports Euro ISDN, NI2, 5ESS or QSIG.
Media Processing Module	Hosting media processing features: conferencing, play/record over HTTP or NFS.
I/O	MOH (Music On Hold), NB (Night Bell).
Ethernet	Dual Redundant 10/100Base-TX Ethernet ports via 2 RJ-45 connectors.
RS-232	Debugging and configuration.
Media Processing	
Voice Coders	G.711, G.726, G.723.1, G.729A, GSM-FR, iLBC, EG.711. Independent dynamic vocoder selection per channel.
Echo Cancellation	G.165 and G.168-2002, with 32, 64 or 128 tail length.
Quality Enhancement	Dynamic programmable jitter buffer, VAD, CNG, 802.1p/Q VLAN tagging, DiffServ,voice quality monitoring, G.729B, RTCPXR.
DTMF/MF Transport	Packet side or PSTN side detection and generation, RFC 2833 compliant DTMF relay. Call Progress tones detection and generation.
IP Transport	VoIP (RTP/RTCP) per IETF RFC 3550 and 3551.
Fax and Modem Transport	T.38 compliant (real time fax), Automatic bypass to PCM or ADPCM.
OSN Server Platform - Embedded, Partner application platform for third-party services	
CPU	▪ OSN1: Intel™ Celeron™ 600 Mhz ▪ OSN2: Intel Pentium M 1.4 GHz

Function	Specification
Memory	▪ OSN1: One SODIMM slot 512M or 1G RAM ▪ OSN2: 1 or 2 GRAM
Storage	▪ OSN1: Single/Dual hard disk drives ▪ OSN2: Single SATA HDD
Interfaces	▪ OSN1: 10/100Base-TX, USB, RS-232, NB relay, MOH ▪ OSN2: 10/100Base-TX, USB, RS-232
Signaling	
Digital –PSTN Protocols	CAS: MF-R1: T1 CAS (E&M, Loop, Start, Feature Group-D, E911CAMA), E1 CAS (R2 MFC), R1.5 numerous protocol and country variants ISDN PRI: ETSI/EURO ISDN, ANSI NI2 and other variants (DMS100, 5ESS) QSIG (Basic and supplementary), IUA (SIGTRAN), VN3, VN4, VN6
Analog Signaling	FXS; Caller ID; polarity reversal; metering tones, distinctive ringing, visual message waiting indication, Loop Start, Ground Start
Control & Management	
Control Protocols	SIP, MSCML
Operations & Management	AudioCodes Element Management System Embedded HTTP Web Server Telnet SNMP V2, V3 Remote configuration and software download via TFTP, HTTP, HTTPS, DHCP and BootP, RADIUS, Syslog (for events, alarms and CDRs) Auto Update
Security	
	IPSec, HTTPS, TLS (SIPS), SSL, Web access list, RADIUS login and SRTP2
Hardware Specifications	
Power Supply	Single universal 90-260 V AC, redundant power supply
Physical	1U high, 19-inch wide
Regulatory Compliance	
Telecommunication Standards	TIA/EIA-IS-968, TBR-4, TBR-13, and TBR-21
Safety and EMC Standards	UL60950-1; FCC 47 CFR part 15 Class B CE Mark (EN55022 Class B, EN60950-1, EN55024, EN300 386, EN61000-3-2/3-3)
Environmental Specifications	ETS 300019-2-1 Storage T1.2, ETS 300019-2-2 Transportation T2.3 ETS 300019-2-3 Operating T3.2

15.2 Mediant 600

The table below lists the main technical specifications of the Mediant 600.

Table 15-2: Mediant 600 Functional Specifications

Function	Specification
Interfaces	
E1/T1/J1	1, 2 or fractional (15 DS0) span spans using RJ-48c connectors
BRI S/T	4 or 8 ports (8/16 calls) using RJ-45 connectors
Analog	4 FXS ports using RJ-11 connectors
Ethernet	Dual Redundant Ethernet 10/100Base-TX Ethernet ports via 2 RJ-45 connectors
RS-232	RS-232 for configuration and troubleshooting
Media Processing	
Voice Coders	G.711, G.726, G.723.1, G.729A, GSM-FR, EG.711, EG.711, iLBC Independent dynamic vocoder selection per channel, VAD, CNG.
Echo Cancellation	G.165 and G.168-2002, with 32, 64 or 128 tail length.
QoS	802.1p/Q VLAN tagging, DiffServ, voice quality monitoring, RTCP-XR DTMF/MF Transport Packet side or PSTN side detection and generation, RFC 2833 compliant DTMF relay, Call Progress tone detection and generation IP Transport, VoIP (RTP/RTCP) per IETF RFC 3550 and 3551.
Fax and Modem Transport	T.38 compliant (real time fax), Automatic bypass to PCM or ADPCM.
Signaling	
E1/T1 CAS	E&M, Loop Start, Feature Group-D, E911CAMA, R1.5, R2 MFC, numerous protocol and country variants
ISDN PRI	ETSI/EURO, ANSI NI2, DMS100, 5ESS, VN3, VN4, VN6 QSIG (Basic Call and Supplementary Services) and other variants, UIA (SIGTRAN)
Control & Management	
Control Protocols	SIP
Operations & Management	AudioCodes Element Management System Embedded HTTP Web Server Telnet SNMP V2/V3
Remote configuration and software download	TFTP, HTTP, HTTPS, DHCP and BootP, RADIUS, Syslog (for events, alarms and CDRs)

Function	Specification
Security	
Security Protocols	IPSec, HTTPS, TLS (SIPS), SSL, Web access list, RADIUS login and SRTP
Hardware Specifications	
Power Supply	Single universal 90-260 V AC power supply
Physical	1U high, 19-inch wide
Dimensions	306 x 273 x 44 mm
Regulatory Compliance	
Telecommunication Standards	TIA/EIA-IS-968, TBR-4, TBR-13, and TBR-21
Safety and EMC Standards	UL60950-1; FCC 47 CFR part 15 Class B CE Mark (EN55022: 2006, EN55024: 1998 + A1: 2001 +A2: 2003 EN6600-3-2: 2000 + A2: 2005, EN6600-3-3: 1995 + A1: 2001 EN60950-1:2001, A11: 2004)
Environmental Specifications	ETS 300019-2-1 Storage T1.2

Reader's Notes

AudioCodes Mediant™ Media Gateways

SIP

Mediant 1000 & Mediant 600

User's Manual

Version 6.0

AudioCodes

www.audiocodes.com